Data-Structure and algorithm:

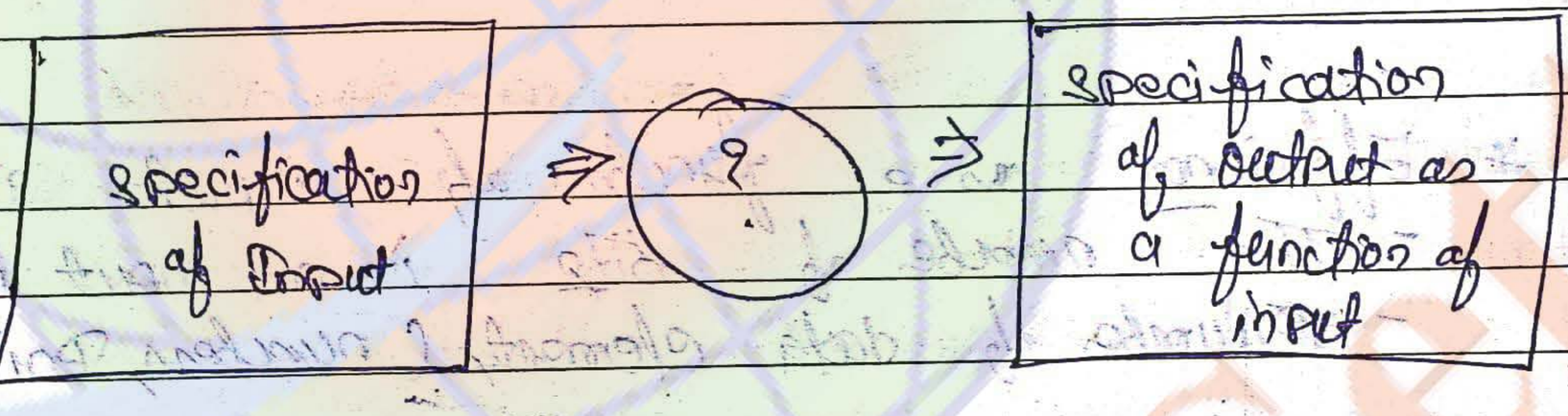
Date __/__/____ Page No.:

(1) Algorithm- outline, the essence of a computational procedure, step-by-step instructs

(2) Program :- An implementation of an algorithm in some Programming language

(3) Data structure:- Organisation of data needed to solve the problem.

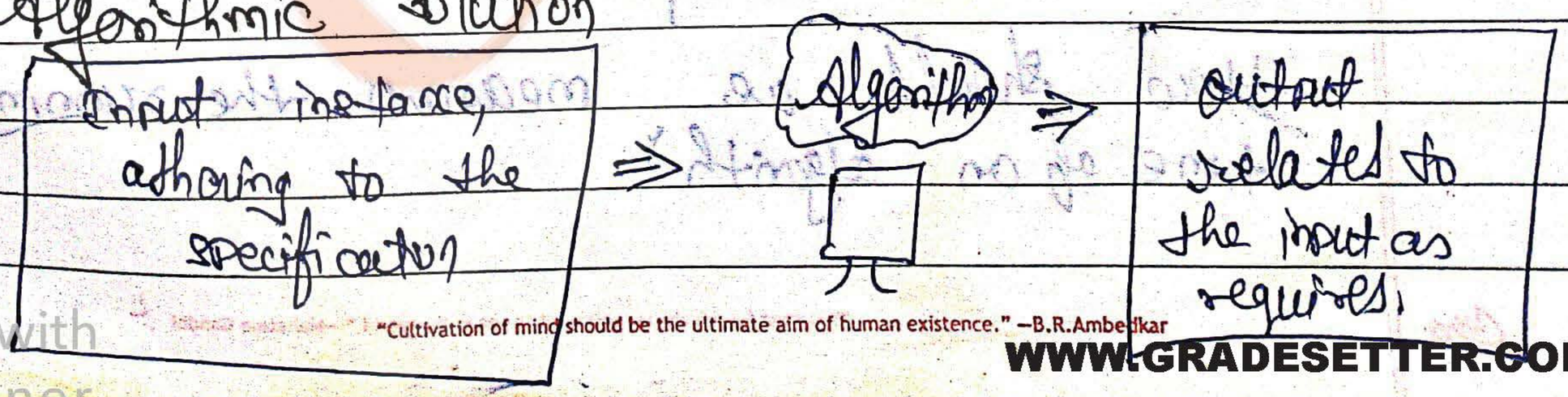
(4) Algorithmic problem:-



- Infinite number of input instances satisfying the specification.
  for eg:- A sorted, non-decreasing sequence of natural number of non-zero, finite length.

  1, 20, 908, 909, 100000, 1000000000
  3

(4) Algorithmic Solution



Date __/__/____ Page No.: ____

- Algorithm describes actions on the input instance.
- Infinitely many correct algorithms for the same algorithmic problem

Q.) What is a good algorithm

- Efficient :-
  - Small running time
  - less space acc.
- Efficiency as a function of input size.

* Efficiency as a function of input size
  - The numbe of bits in an input number
  - Number of data elements (numbers, Points)

Q.) measuring the running time!

How should we measure the running time of an algorithm?

Date __/__/____ Page No.: ____

- Experimental study!-
  - write a program that implements the algorithm
  - Run the program with data sets of varying size and composition.
  - Use a method like "System.currentTimeMillis()" to get an accurate measure of the actual running time.

⇒ Limitations of experimental studies:-

- It is necessary to implement and test the algorithm in order to determine it's running time.
- Experiments can be done only on a limited set of inputs, and may not be indicative of the running time on another input not included in the experiment.
- In order to compare two algorithms, the same hardware and software environments should be used.

P.) Beyond experimental studies:-

We will develop a general methodology for analyzing

BMW का founder QS को हैं। जो problem की आठ वाट कर काम करता है।

BMQS

(1)

Divide and Conquer	Greedy :-
- Binary sort	- Travelling salesman Problem
- merge sort	- Knapsack Problem
- Quick short	- Kruskals algo } minimal spanning tree.
- Strassen's matrix multiplication	- Prim's algo.
	- Job scheduling model.

(2) Complexity :-

methods	Time complexity Best	Avg.	worst	space complexity		comment
• Binary search						const $O(\log n)$
merge sort	$O(n \log(n))$	$O(n \log(n))$	$O(n \log(n))$	$O(n)$	✓	
Quick sort	,,	,,	$O(n^2)$	,,	depends	
strassen matrix						$O(n^2)$
• minimal spanning :-						~~$O(E \log_2 E)$~~
Kruskal algo						$O(E \log_2 E)$
Prims algo						$O(E \log_2 V)$ const

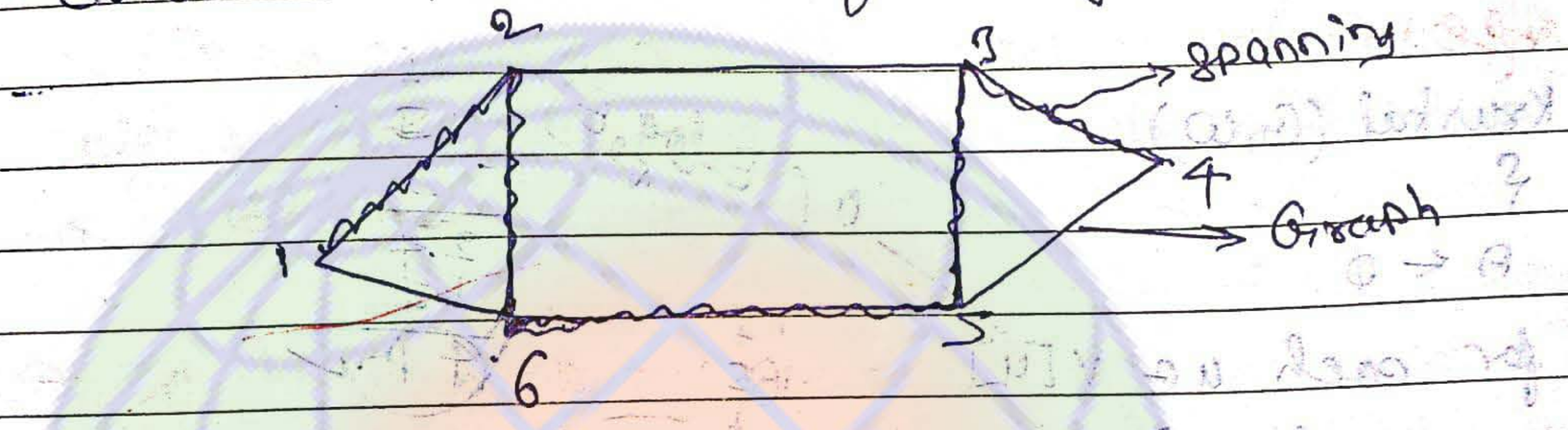
WWW.GRADESETTER.COM

★ Date Minimal Spanning tree:- (min marks = 2 weightage) Page No.

Tree :→ A tree is a connected acylic graph.

• Spanning tree:-

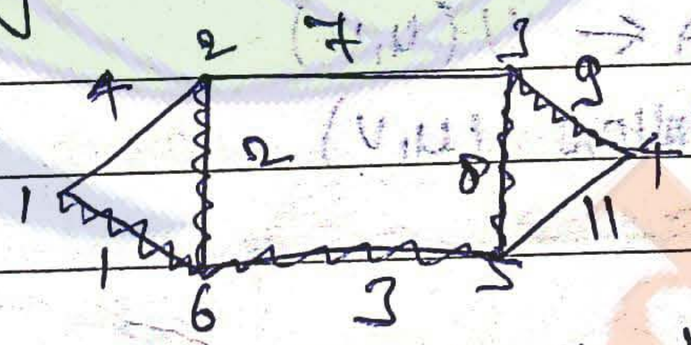
(*) A spanning tree T is a subgraph of a Graph G, such that first it is a tree, 2nd it must cover all the vertices of the graph.



• minimal spanning tree:-

A spanning tree of a weighted graph is said to be minimal if it has the minimal optimal weight.

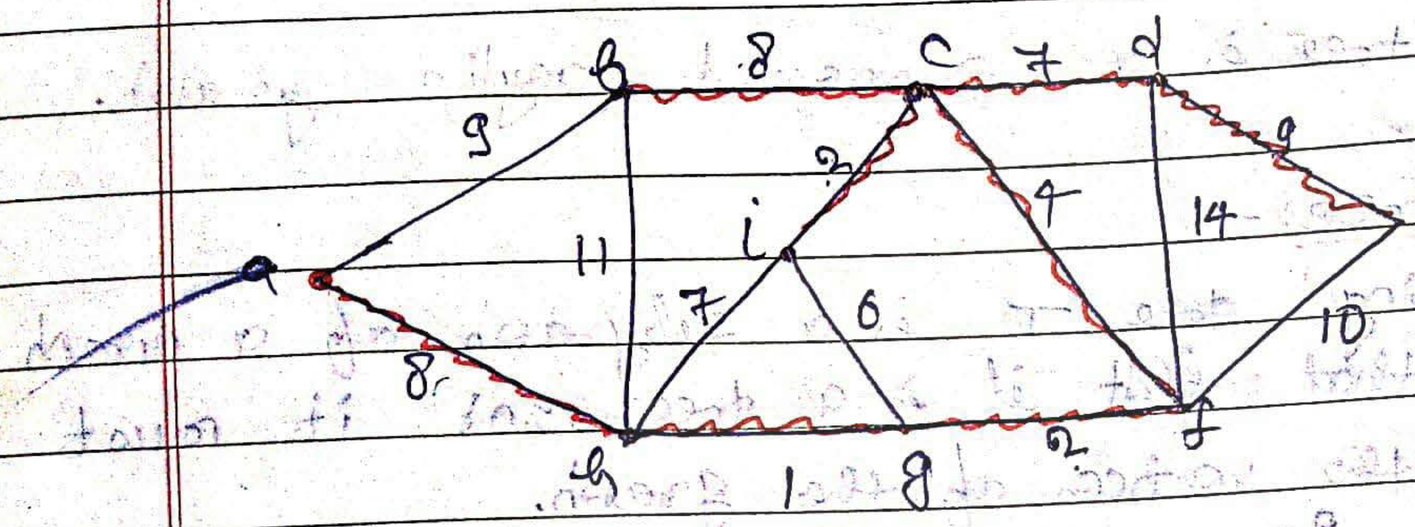
Note:- But uniquness is not guarantee and, we may have some other spanning tree which have the same weight.



(1) Kruskal algorithm:- {Take the lowest value first}

a) Consider a weighted Undirected graph and find a minimal spanning tree using kruskal's algorithm.

Date __/__/____ Page No.: ____



Ans = 41.

Algo:-

Kruskal (G, w)

{

$A \leftarrow \phi$

for each $u \in V[G]$

makeset(u)

sort the edge $(u,v) \in E(G)$ in non-decreasing order

for each edge $(u,v) \in E(G)$ taken in non-decreasing order

{

if {find set(u) $\neq$ Find set(v)}

{

$A \leftarrow U(u,v)$

UNION (u,v)

}

}

return A

$O(E \log_2 E)$

Note:-

makeset(u) :- makeset is a function which is when called on a vertex v, makes it a tree or convert it into a tree.

Find set of v :- find set of v is a function which is when called on a vertex, will give the

set of all the vertex, which are rechable from U.

C :- we check this condition c, because we find the vertex u and v already connected or not.

Because if yes, then the inclusion of the edge "u, v" will lead to a cycle. and that will be a problem.

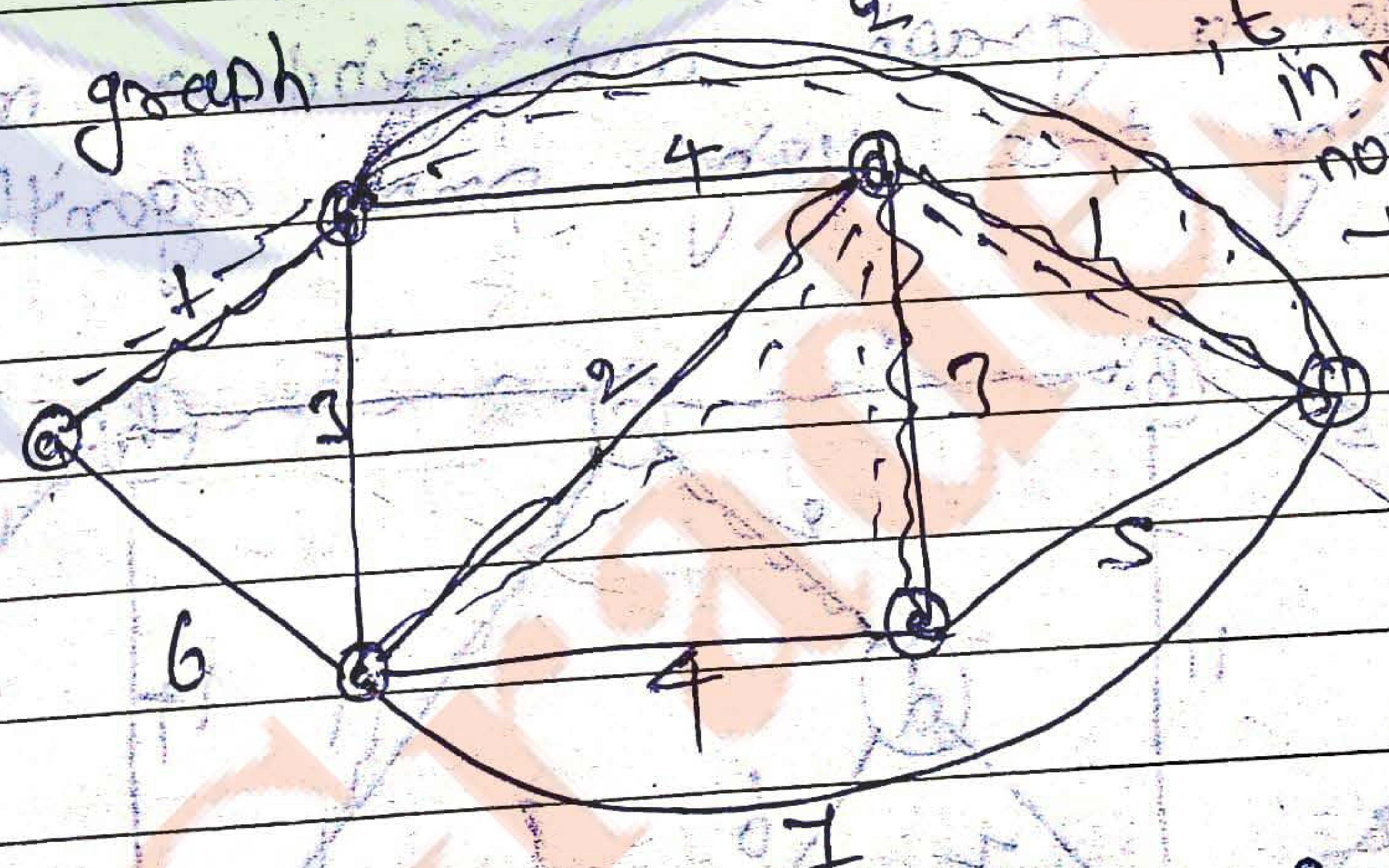
* Important Point :-
* Kruskal works on a purely greedy approach being greedy about weight.

* It guarantee to give optimal soln (Uniqueness is not guarantee)

* The cost of $O(E \log_2 E)$

Methods:- (Note:) The methods of solving of this type of problem is to start with the minimum value, and in the sequence you increase it to keep the thing in mind that there is no cycle formation takes place.

Q1.) Consider a graph



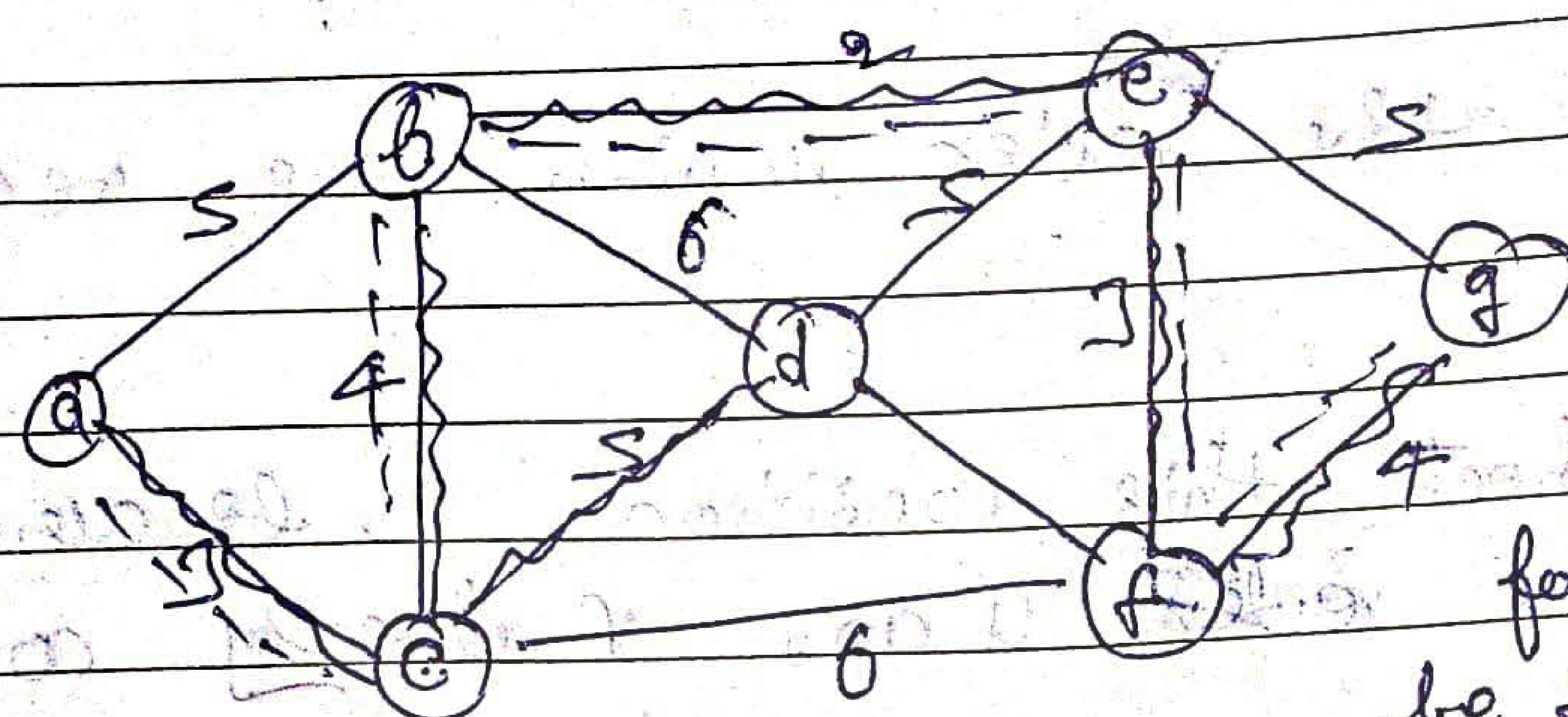
which of the following can not be the sequence of edges added if kruskal is used.

ab, bf

(a) ab df bf dc de
(b) ab df dc bf de
(c) df ab dc bf de
(d) df ab bf de dc → This cannot be allowed.

Chitra

Q2)



which of the following can not be sequence in kruskals algorithm:-

(a) $\overset{2}{be}$ $\overset{3}{ef}$ $\overset{3}{ac}$ $\overset{4}{bc}$ $\overset{4}{fg}$ $\overset{5}{ed}$ ✓

(b) $\overset{2}{be}$ $\overset{3}{ef}$ $\overset{5}{ac}$ $\overset{4}{fg}$ $\overset{4}{bc}$ $\overset{5}{cd}$ ✓

(c) be ac ef bc fg ed

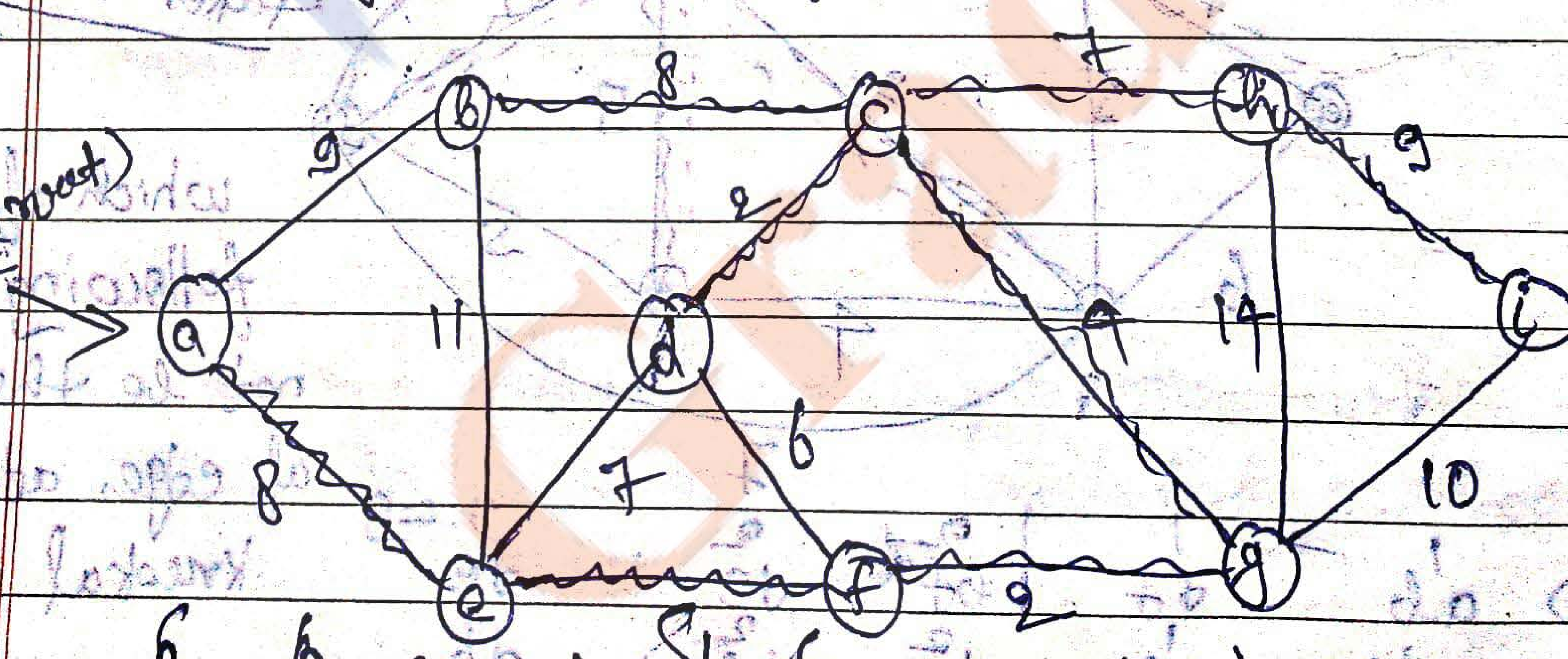
X (d) $\overset{2}{be}$ $\overset{3}{ef}$ $\overset{4}{bc}$ ac fg cd

↑ we have left 3, so we not take 4 before 3.

★ Prim's algorithm:-

{ • start from root node, which you assume

Q) Consider a graph and find a minimal spanning tree using prims algorithm



	a	b	c	d	e	f	g	h	i	
parent π(u)	–	a	–	e	a	e	–	–	–	
distance d(u)		9	∞	7	1	1	∞	∞	∞	

Date __/__/____ Page No.: _____

```
Prims (G, w, r)
{
  for each u ∈ V(G)
    {
      π[u] ← ϕ
      d[u] ← ∞
    }
  d[r] ← 0
  Q ← V[G]
  while Q ≠ ϕ
    u ← delete min(Q)
    for each v ∈ adj(u)
      if (v ∈ Q) && (w(u,v) < d[v])
        {
          d[v] ← w(u,v)
          π[v] ← u
        }
    }
}
```

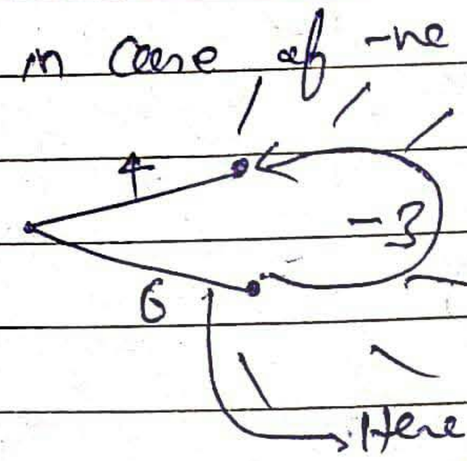
u = e

a, b, f, d

↳ Here, we first take all the distance as infinity(∞) and by using parity queue, we apply the minimum distance as then after, it select it. Do the same process for each of the node.

Date __/__/____ Page No.: ____

Note:⇒ Fails in case of -ve



Here due to -ve value we not get better with 6, but we not know in initial so, prims fails for -ve values.

✱ ⇒ Important point:-

(i) Prim algorithm also work on greedy approach.

(ii) Prims also guarantee to give optimal solution

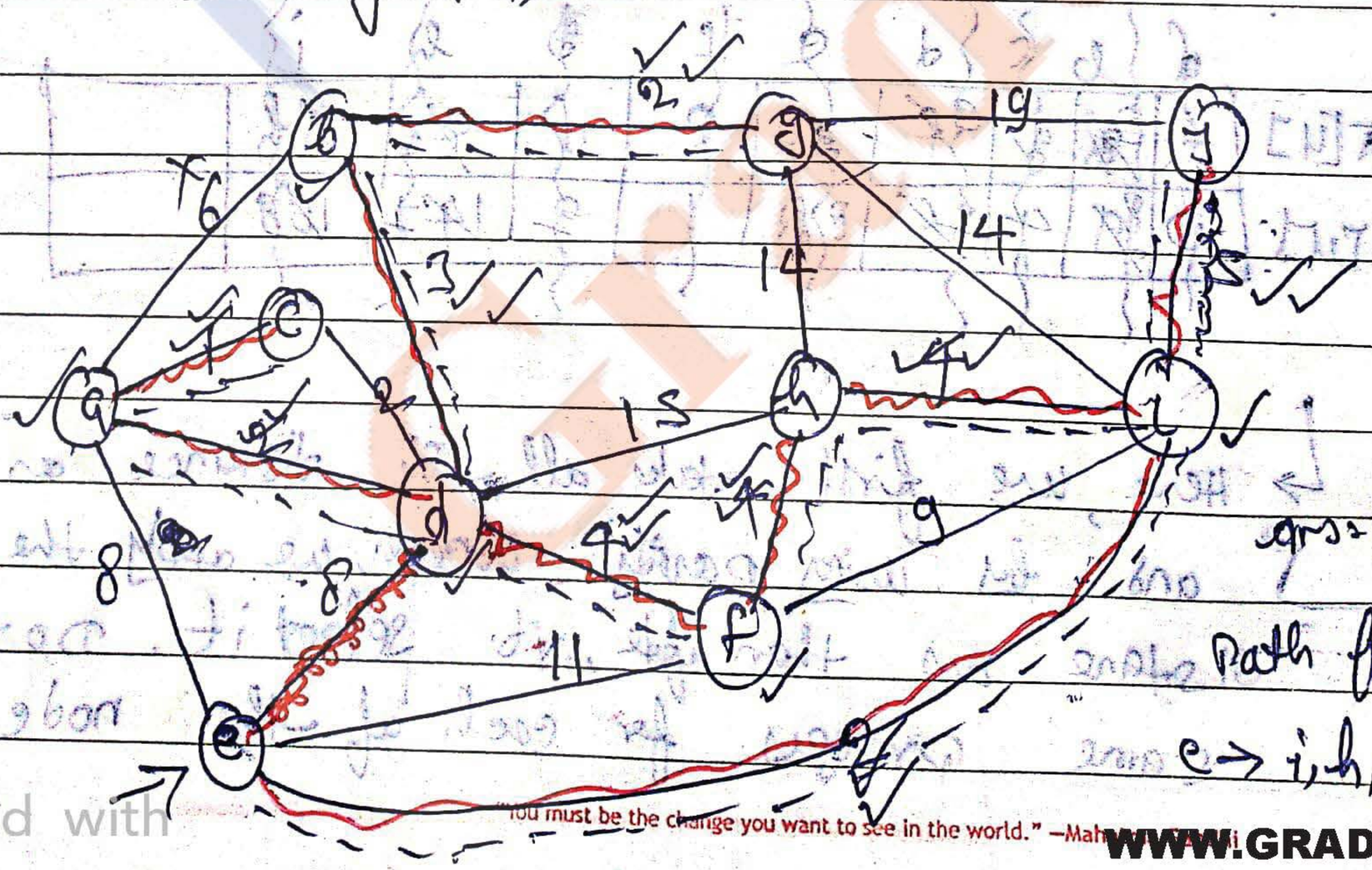
(iii) Uniqueness is not guarantee, by running Prims on diffb. root or diffb starting vertex, we may have diffb spanning tree, but the cost will be same.

(iv) On negative(-ve) weight Prims may fails because in that case local optima may or may not be global optima.

1+

2007 Q:) Consider a graph,

27



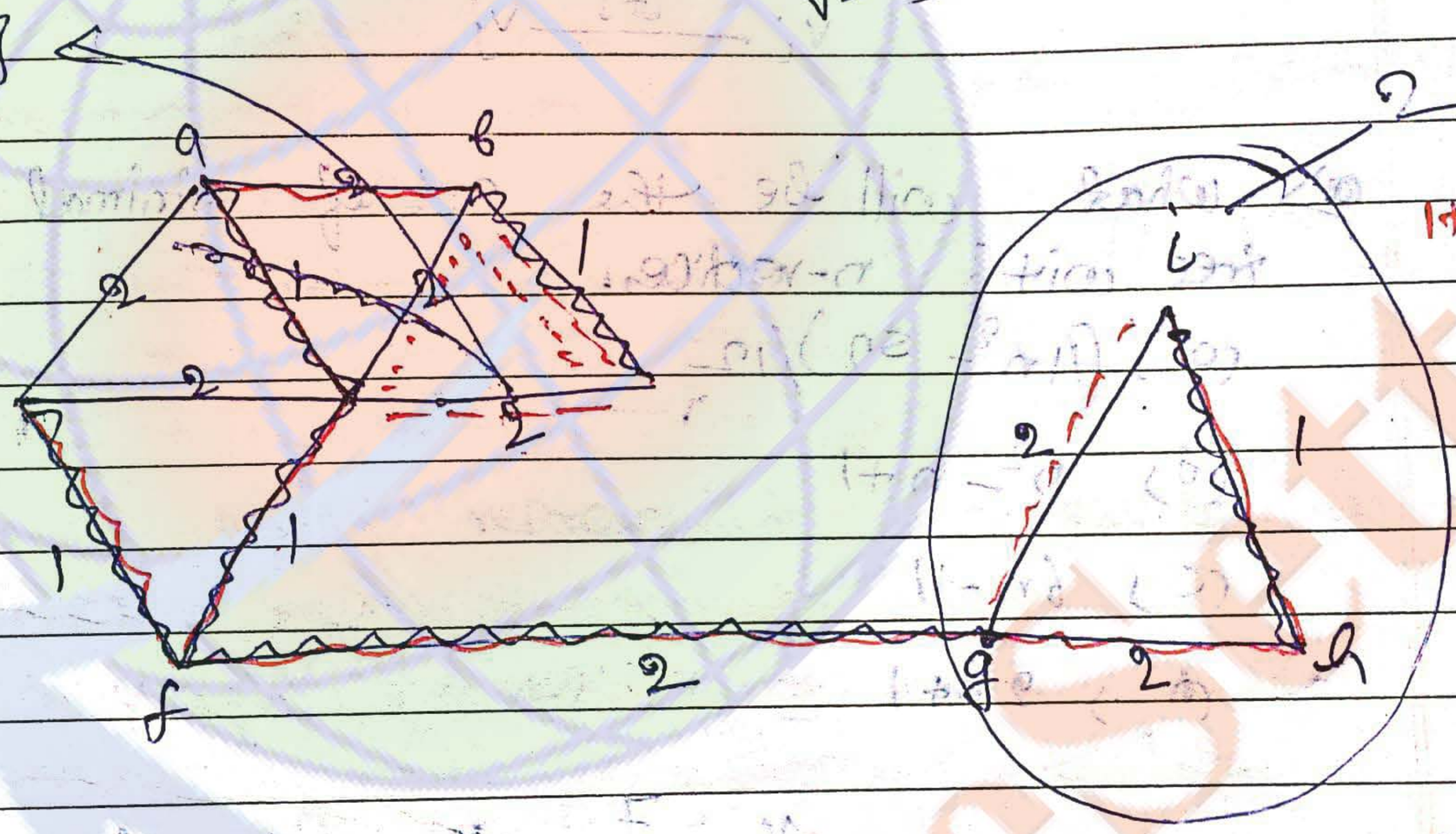
Path from e:-

e → i, h, f, d, a, c, b, g, j

Date __/__/____ Page No.: _____

- If the cost or the weight of every edge is distinct then the tree will be unique. irrespective of the algo or the starting vertex used.
- If we have two different ~~minimal~~ minimal spanning tree in a graph then it implies atleast two edges with the same weight.
- In the best case, the complexity of Prims will be $O(E \log_2 V)$

2014 Q.7



How many distinct/different minimal tree are possible

(a) 4 (b) 5 (c) 6 (d) 7

Sol/n $3 \times 2 = 6$ //

Date __/__/____ Page No.: ____

2011

a) Consider a Undirected graph, with n-number of vertices like this

$v_1, v_2 \text{ ---} v_n$

there is an edge b/w two vertices v_i and v_j if and only if

$0 < |i-j| \leq 2$

and the cost or the weight of the edge v_i or v_j is $i+j$

$v_i \xrightarrow{i+j} v_j$

q) what will be the cost of minimal spanning tree with n-vertices.

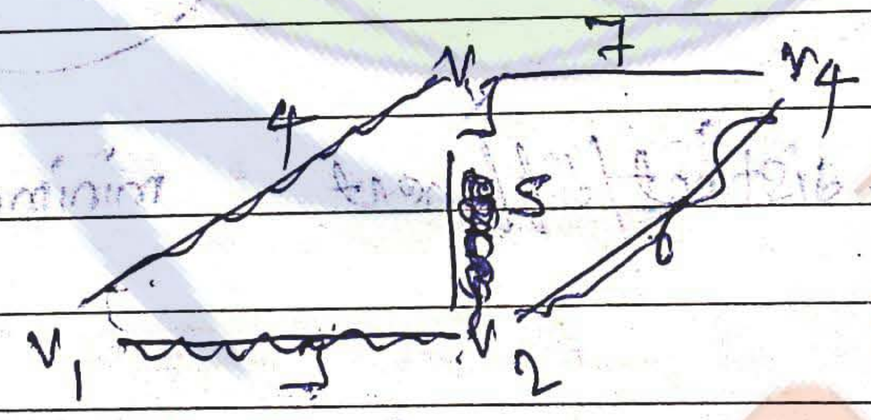
(a) $(11n^2 - 5n)/12$

(b) $n^2 - n + 1$

(c) $6n - 11$

(d) $2n + 1$

Sol n



Check for n = 4

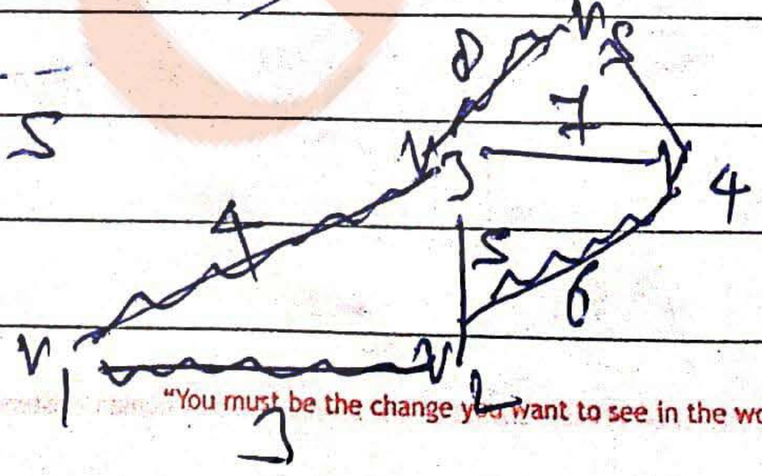
n = 4

cost = 13

$(10 \times 11 - 5 \times 4)/12$

$(16 - 4 + 12 \cdot 13)$ b ✓

check for n = 5



n ≥ 5

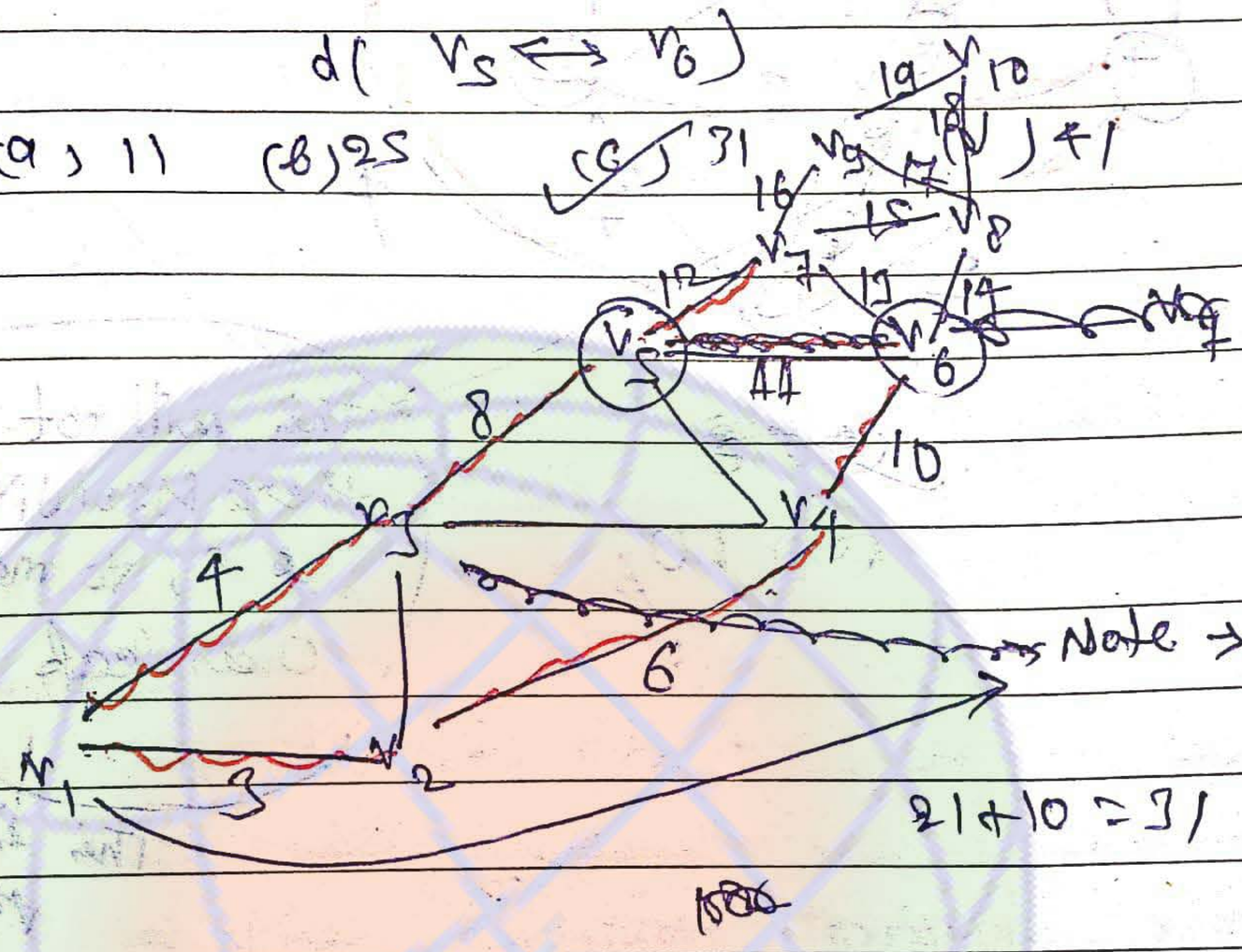
a ≥ b ✓

(b) Imagine a graph from v_1 to v_{10}, what ei the distance b/w v_5 to v_6.

$d(v_5 \leftrightarrow v_6)$

(a) 11 (b) 25 (c) 31 (d) +1

So/n



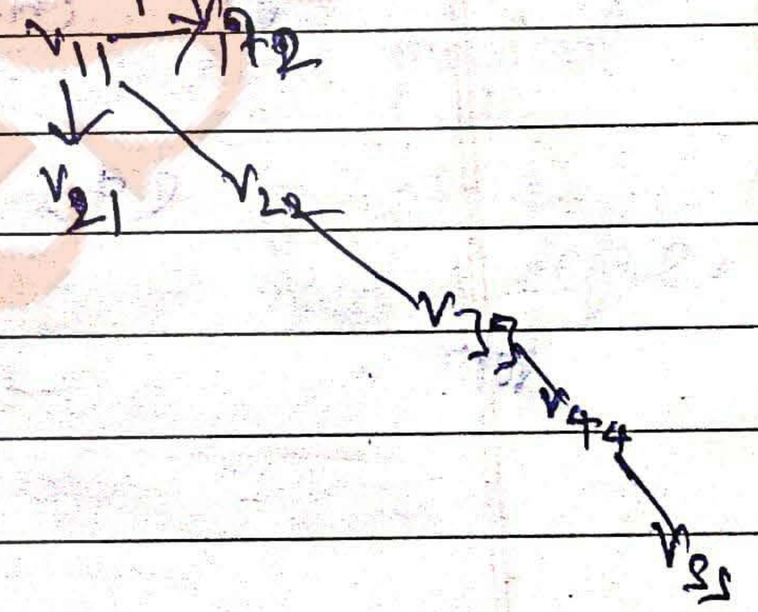
2010

a) Consider a matrix representation of a graph having 5 vertex.

$V = \{0, 1, 2, 3, 4\}$

$W =$

	0	1	2	3	4
0	0	1	8	1	4
1	1	0	12	4	9
2	8	12	0	7	3
3	1	4	7	0	2
4	4	9	3	2	0



(a) find the minimum possible weight of spanning tree. such that the vertex 0 must be the leaf vertex

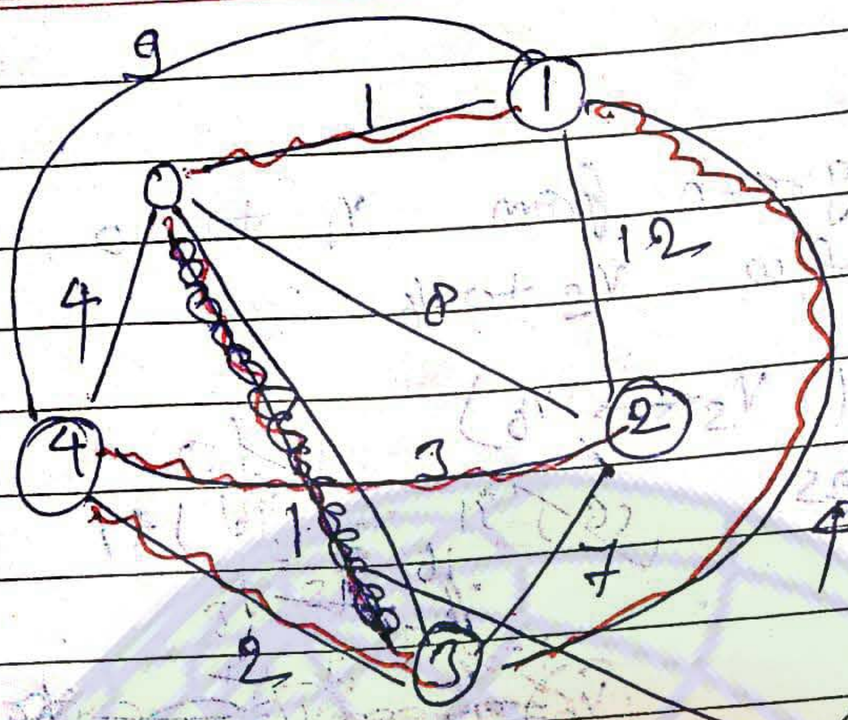
(a) 7 (b) 8 (c) 9 (d) 10

C → 2 cab/

Page No.:

Date __/__/____

Soln



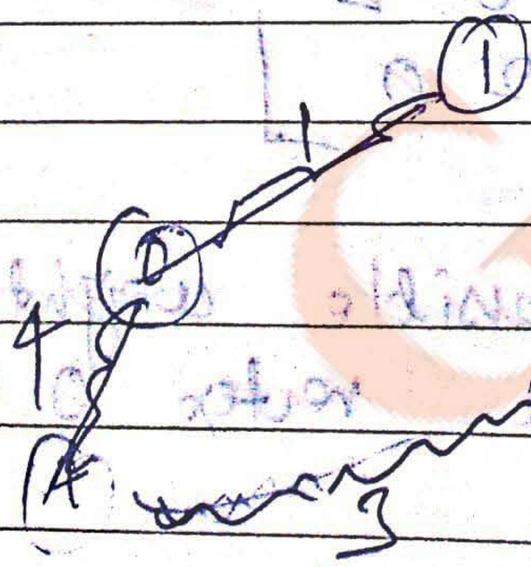
(d) 10 //

This will not, because there M condition in ↓ 0, to make 0 or leaf.

This is the trick of this question //

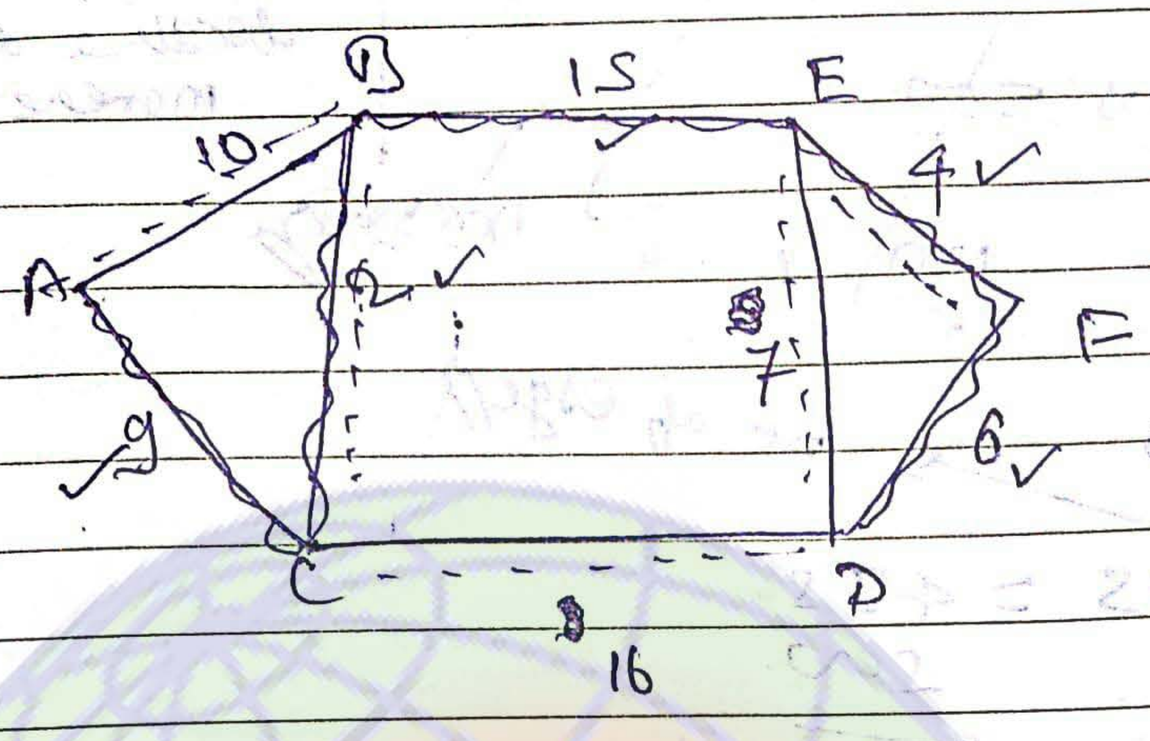
(b) What will be the minimum possible weight of a path P, from vertex 1 to vertex 2 it can contain at most 3 edges //.

(a) 7 (b) 6 (c) 9 (d) 10



Date __/__/____ Page No.: _____

2015
Q> Consider a graph,



minimum = 36 (Ans)

Contain edge:-
AC, BC, BE, EF, DF

what is the minimum possible sum of the weight of all edges in the graph.

fill in the blank

Sol'n

1+2+3 → Not possible because minimal tree of the graph is decides before.

Ans ≯ 29

2015
Q> Consider a undirected graph G, with 100 vertex and 500 edges. and the weight of minum spanning tree is 500. If by mistake the weight of each edge is increased by 5. find the ~~decrease~~ weight of ~~maximum~~ minimum spanning tree.

"Education of mind should be the ultimate aim of human existence." —B.R. Ambedkar

Date ___/___/______

100 → V = 1
300

sum = 500

100 | 1

1×1 3 = (9) //

| minimal spanning tree is same because all is increase same

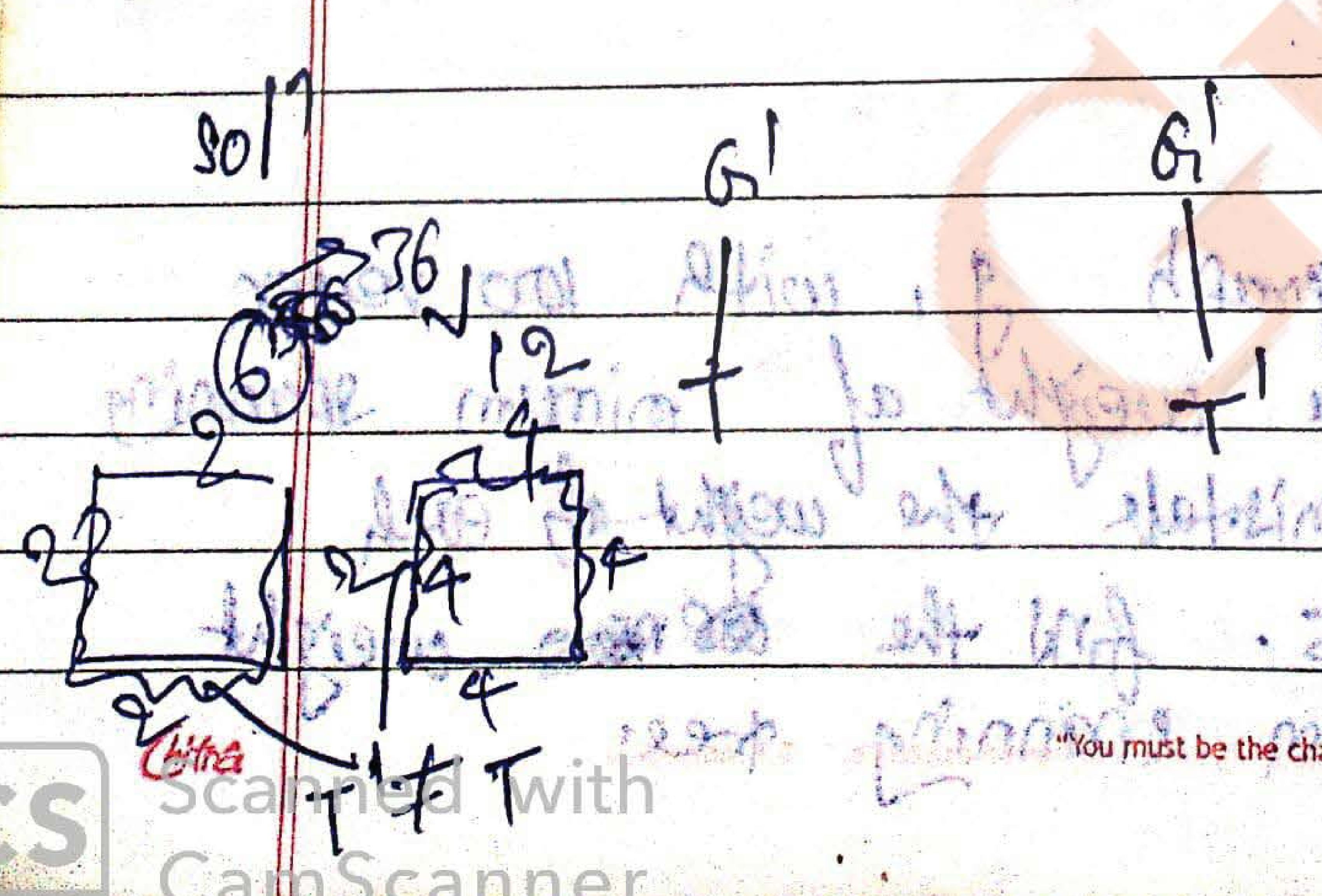
100 → no of edge //

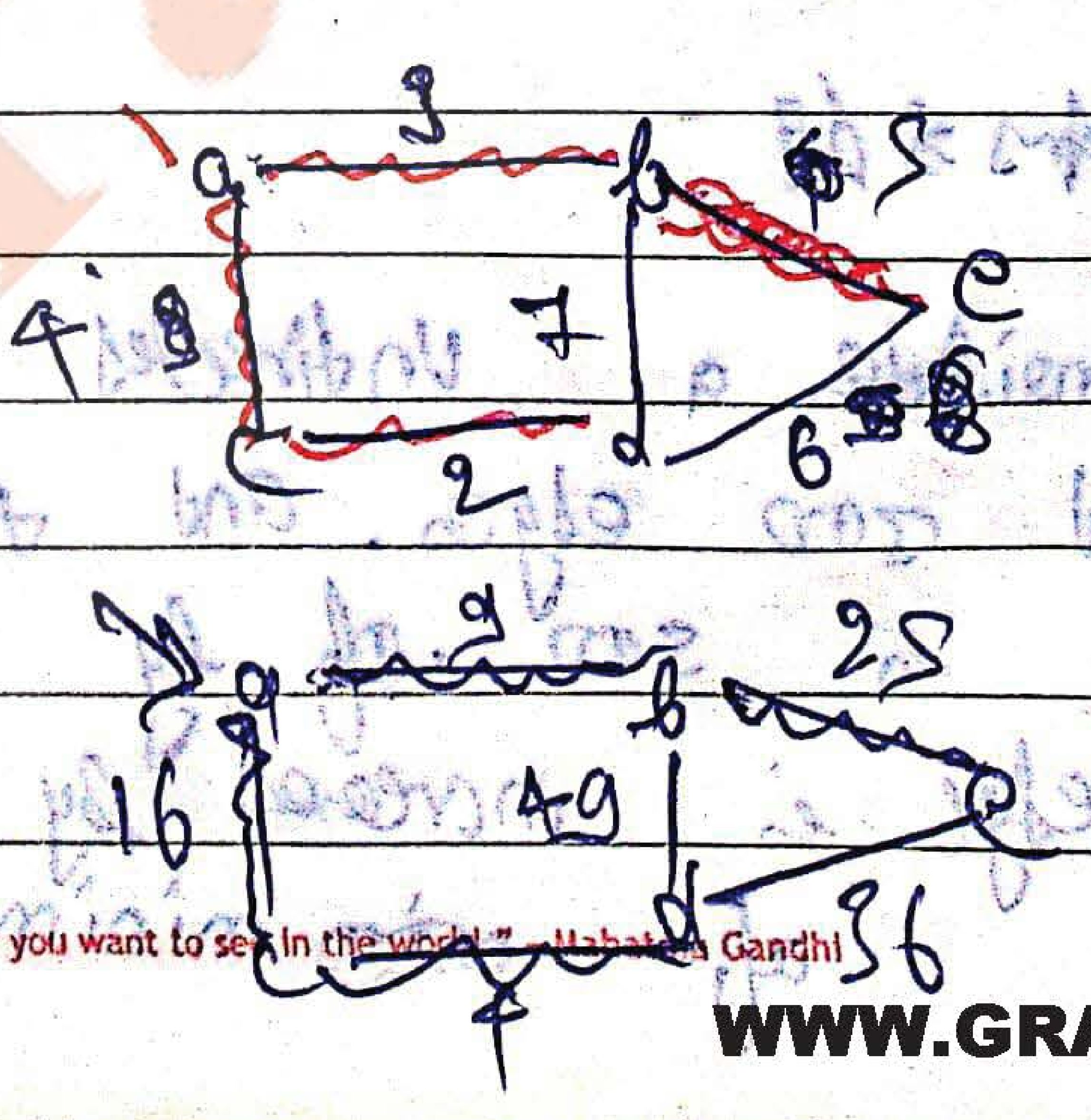
99 × 5 = 495
500
995 A

Q.> Consider a graph G, and G' such that G' is constructed by squaring the edges which was in g.
(weight greater than 1)
Let T and T' be the minimal spanning tree on G and G'.
which of the following is true

(A) T' = T | t' = t², t ← weight
(B) T' = T | t' < t²
(C) T' ≠ T | t' = t²
✓(D) null

Sol^n



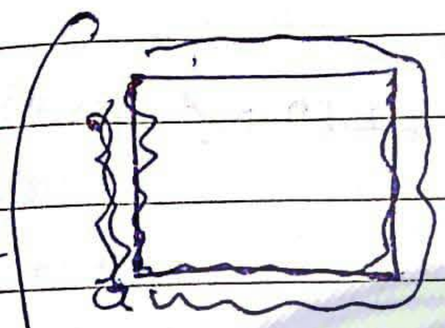


Date __/__/____ Page No.: ____

Q:> What is the largest integer m, such that every simple connected graph on n-vertices and n-edges contains atleast m distinct spanning trees.

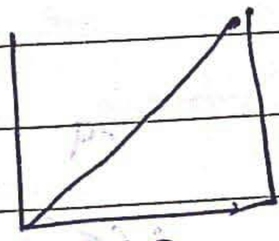
(a) 1 (b) 2
(c) 3 (d) n

Soln



n=4 ⇒ n=4
v=4
e=4

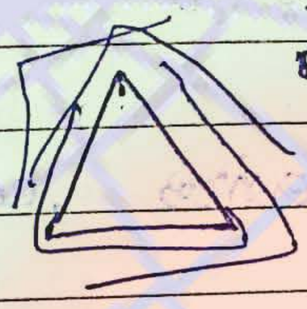
n=4

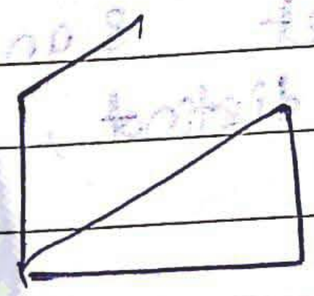


3

m=5

n=3 2
m=3



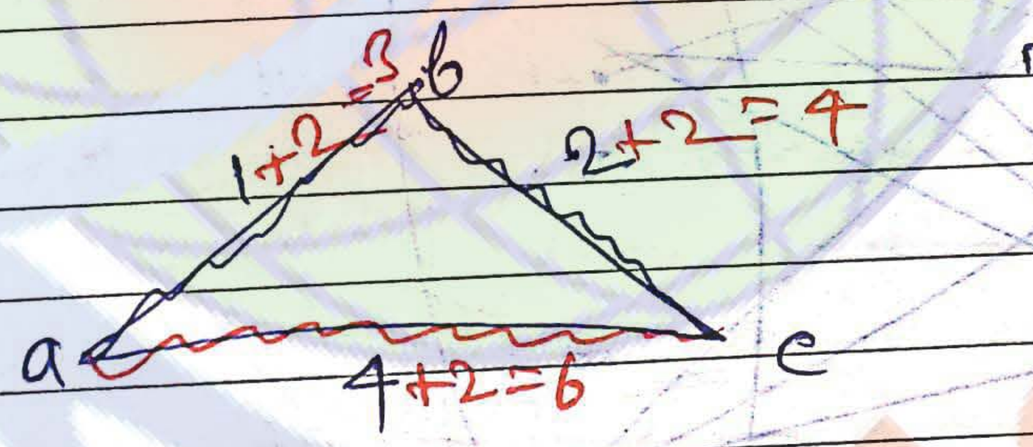


Ques 2

to we need to find the minimum to three.

Solⁿ

↳



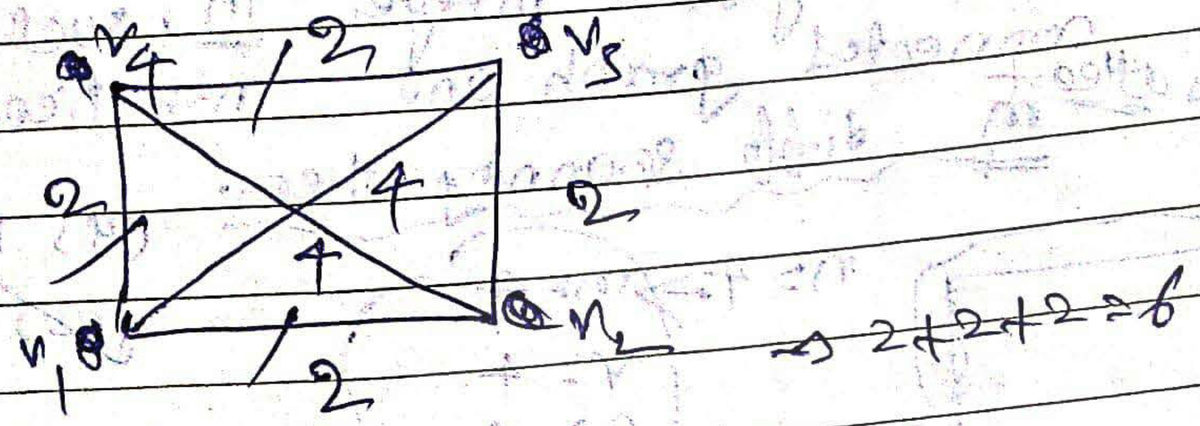
If we add the same value of distinct edge value then there may be possibility that path will change

Q> Consider a weighted complete graph G, such that weight of edge v_i, v_j will be $2|i-j|$

$$w(v_i, v_j) = 2|i-j|$$

what will be the weight of minimum spanning tree.

(a) n-1 (b) 2n-2 (c) n (d) 2

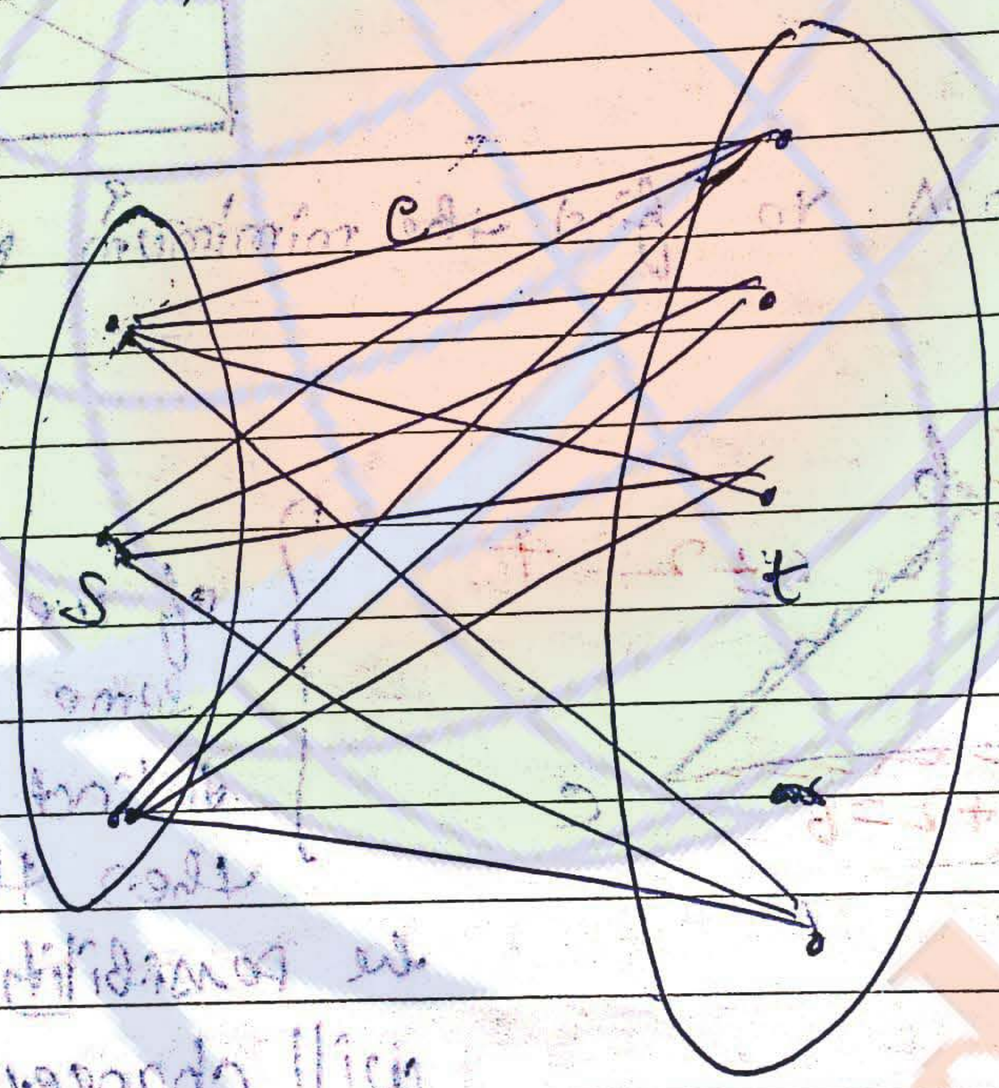


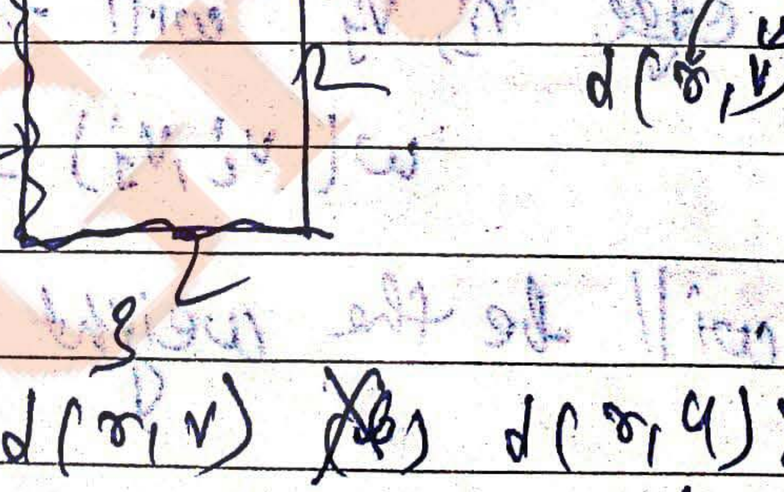
$\Rightarrow 2+2+2=6$

(a) $4-1$ ✗

(b) $4\times2-2 \Rightarrow 8-2\geq6$ ✓

Q.) Let S and T be two vertices, each having distinct,





(a) $d(x,y) < d(x,v)$ (b) $d(x,y) > d(x,v)$

(c) $d(x,y) \leq d(x,v)$ (d) none of all

Date __/__/____ Page No.: ______

★ Single source shortest Path:-

↳ min mark = 2

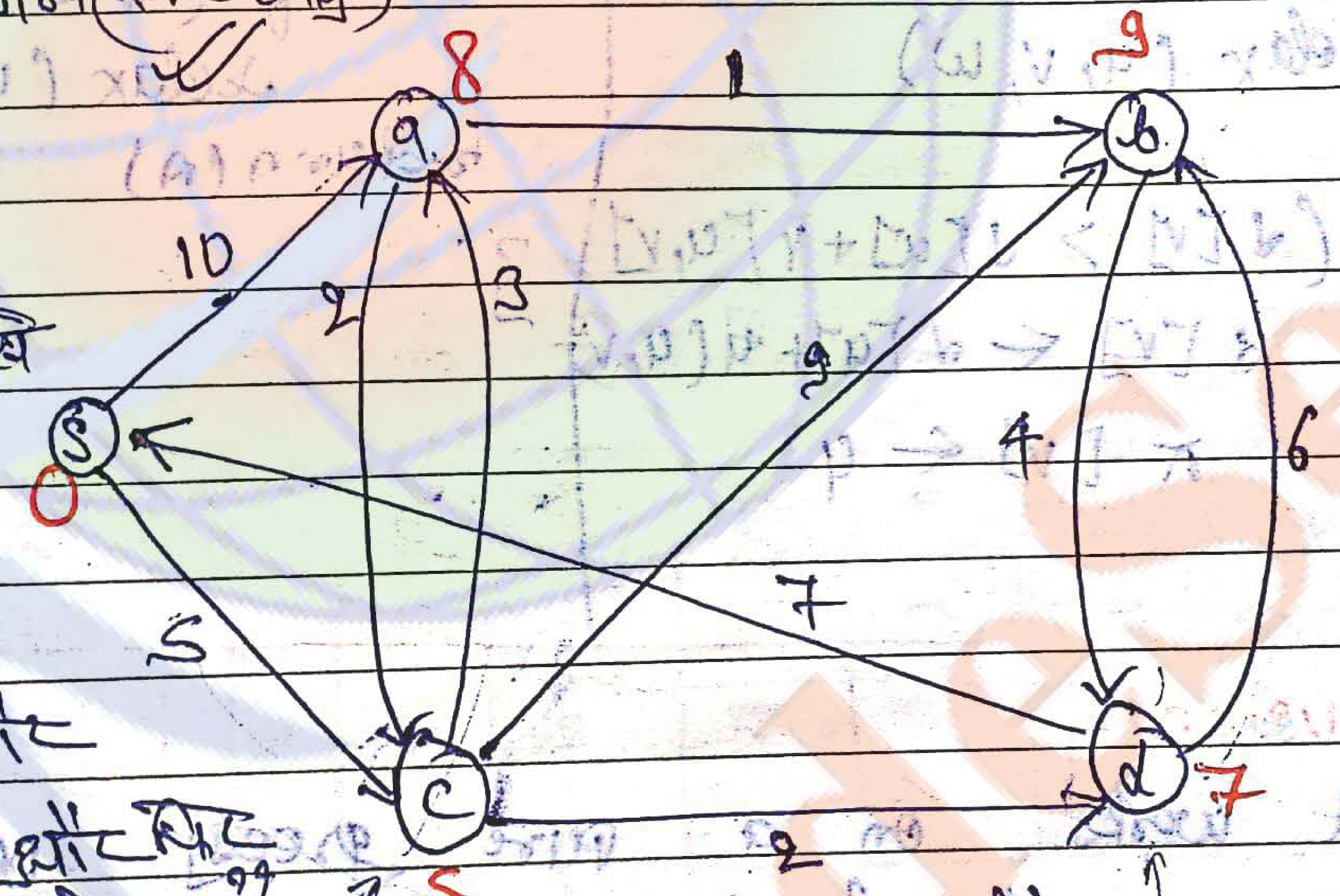
Q) Consider a graph G, which is a weighted directed graph, the problem is from any vertex called source vertex, we have to find the shortest Path to all the other vertex of the graph.

weight in general can be (+)ve as well as (−)ve.

(i) Dijkastra algorithm:-

Dijkastra (+ve only):-

इसमें हम source से सारे point का distance लिखते जाएंगे कितना possible है, और फिर इसमें से "shortest" निकाल इसमें जितना "सबसे कम distance" होगा, उसको vertex मानकर आगे find करेंगे



	s	a	b	c	d
π[u]	–	c	a	s	c
d[u]	0	8	9	5	7

a(8) ⟸ c ⟸ s

b(9) ⟸ a ⟸ c ⟸ s

c(5) ⟸ s

d(7) ⟸ c ⟸ s

Date __/__/____ Page No.: ____

Algo:-

~~Dijkastra (G,w,s)~~

```
Initialize single source(G,s)
{
for each v∈ V(s)
  {
    π[v] ← ϕ
    d[v] ← ∞
  }
d[s] ← 0
}

Relax (u, v, w)
{
if (d[v] > d[u] + r[u,v]
  d[v] ⇐ d[u] + u(u,v)
  π[v] ⇐ u
```

```
dijkastra (G, w, s)
{
initialize single source (G,s)
A ← ϕ
Q ← V[G]
while Q ≠ ϕ
  u ← delete min(Q)
  A ← A ∪ [u]
for each v ∈ adj[u]
  if (u ∈ Q)
    relax (u, v, w)
return (A)
}
```

Conclusion:-

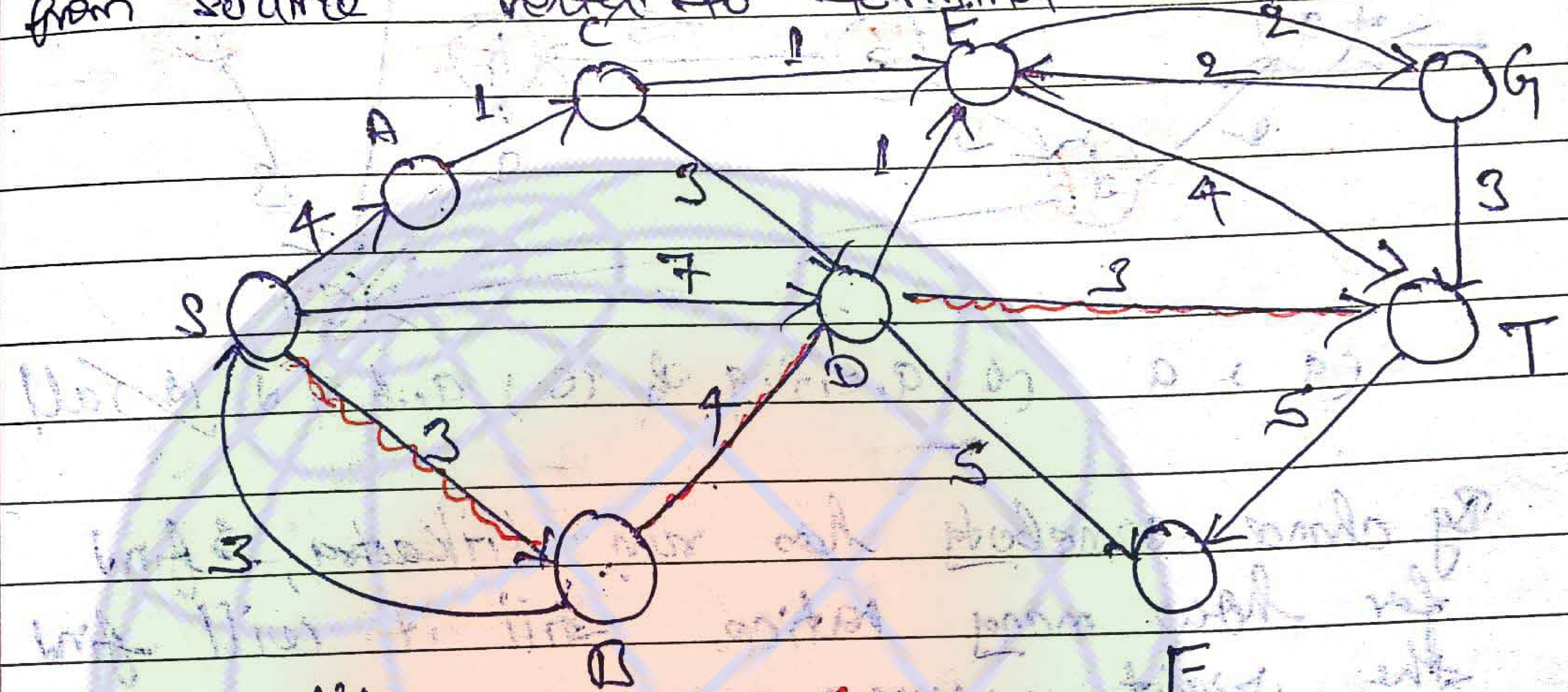
(i) It works on a pure greedy approach being greedy about waght.

(ii) It guarantees to give minimal cost path from every vertex from the source vertex.

(iii) Dijkastra may fail with -ve weight edges.

(iv) minimum cost $O(V \cdot \log_2 V)$

Q) consider a graph and find the minimum distance, from source vertex to terminal vertex.



Dijkstra path:-

(a) S D T
(b) S B D
(c) S A C D T
✓(d) S A C E T

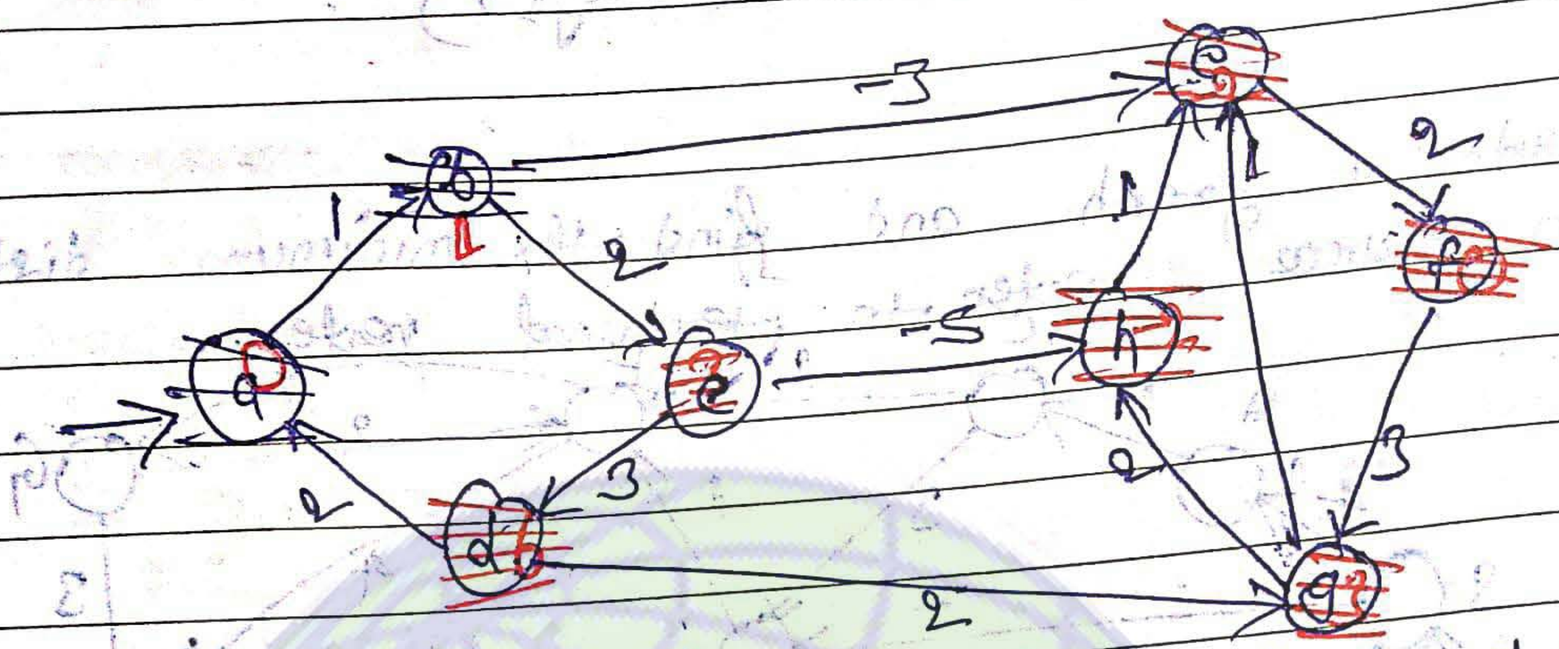
	S	A	B	C	D	E	F	G	T
π[u]	-	A	S	-	B	D	D	-	D
d[u]	0	4	3 ∞	∞	4	∞ 1	∞ 5	∞	∞ 3

	S	A	B	C	D	E	F	G	T
π[u]	-	S	S	A	S	C	D	E	E
d[u]	0	4	3	5	7	6	12	8	10

S → S → C → E → T

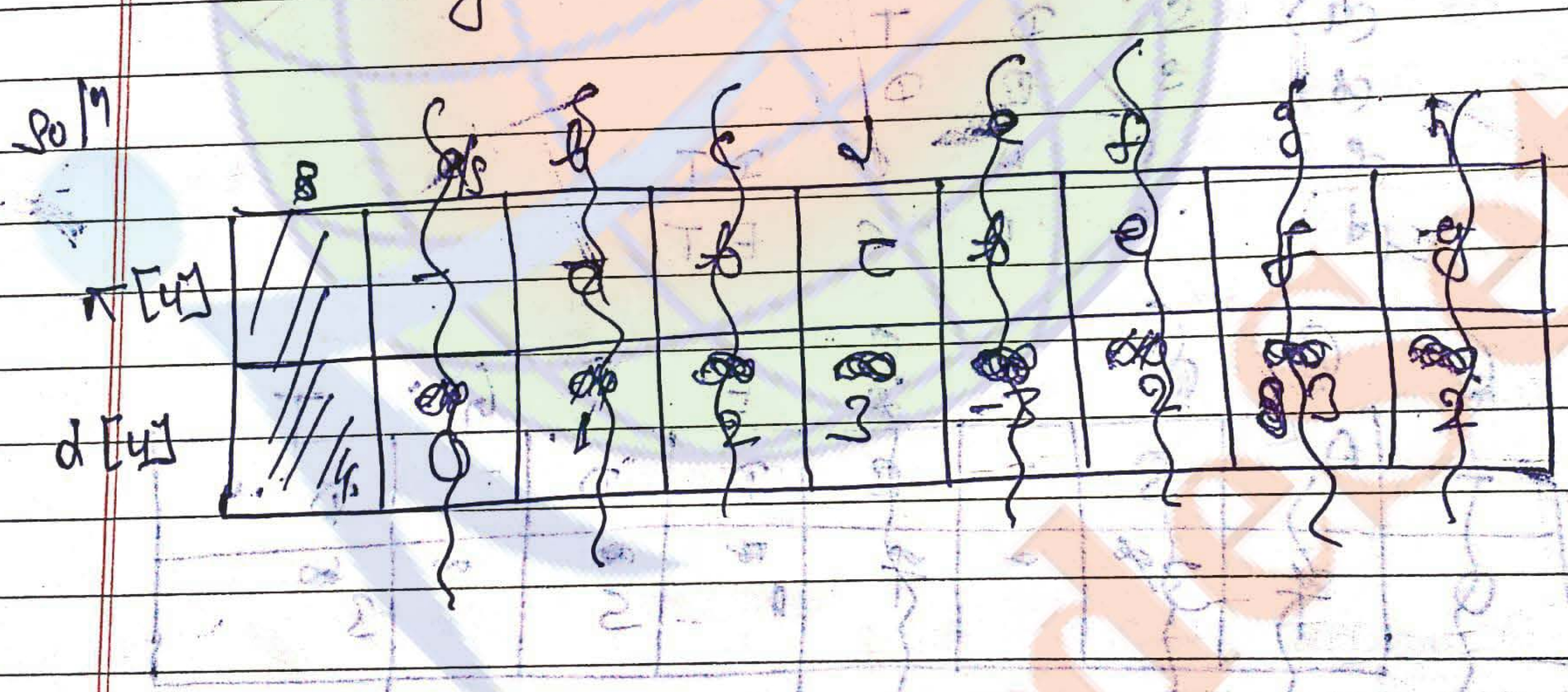
Date __/__/____ Page No.: ____

Q) Consider a graph



(a) a (b) a, e, f, g, h (c) a, b, c, d (d) all

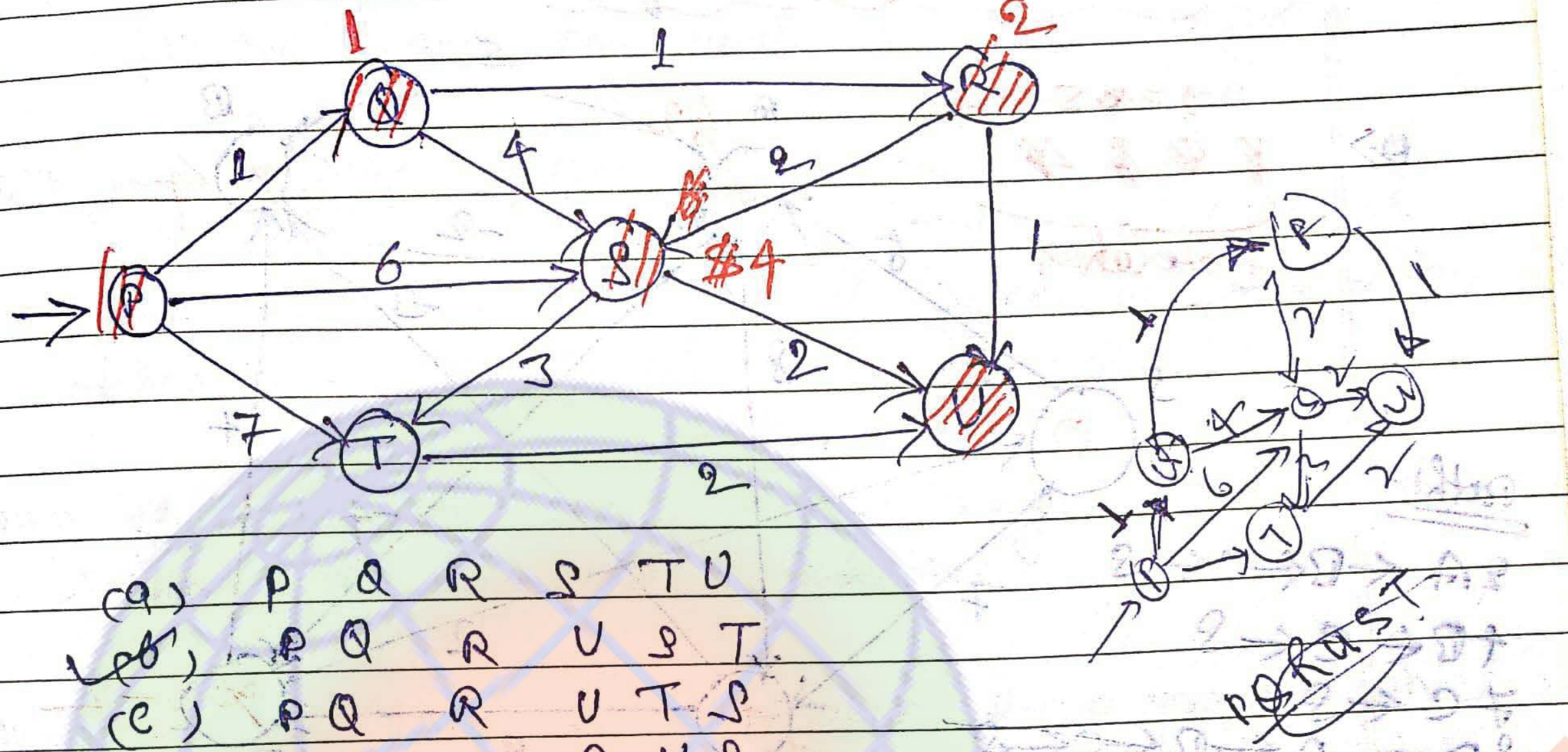
By chance somebody has run dijkastra, & found for how many nodes still it will find the right answer.



Hint) → To check the shortest path after each relax.

Date __/__/____ Page No.: _____

Dear Gear Q. Consider a graph



(a) P Q R S T U
(b) P Q R U S T
(c) P Q R U T S
(D) P Q T R U S

In what order, node get finalize if dijkastra will ~~allows~~ start from P.

So/n

	P	Q	R	S	T	U
π[u]	—	P	Q	R	P	R
d[u]	0	1	1	4	7	2

P — Q — R — U — S — T

Note:-

(*) The problem with Dijkastra algorithm is that it fails on -ve weight edges

⇓

Solution is

⇓

Bellman form

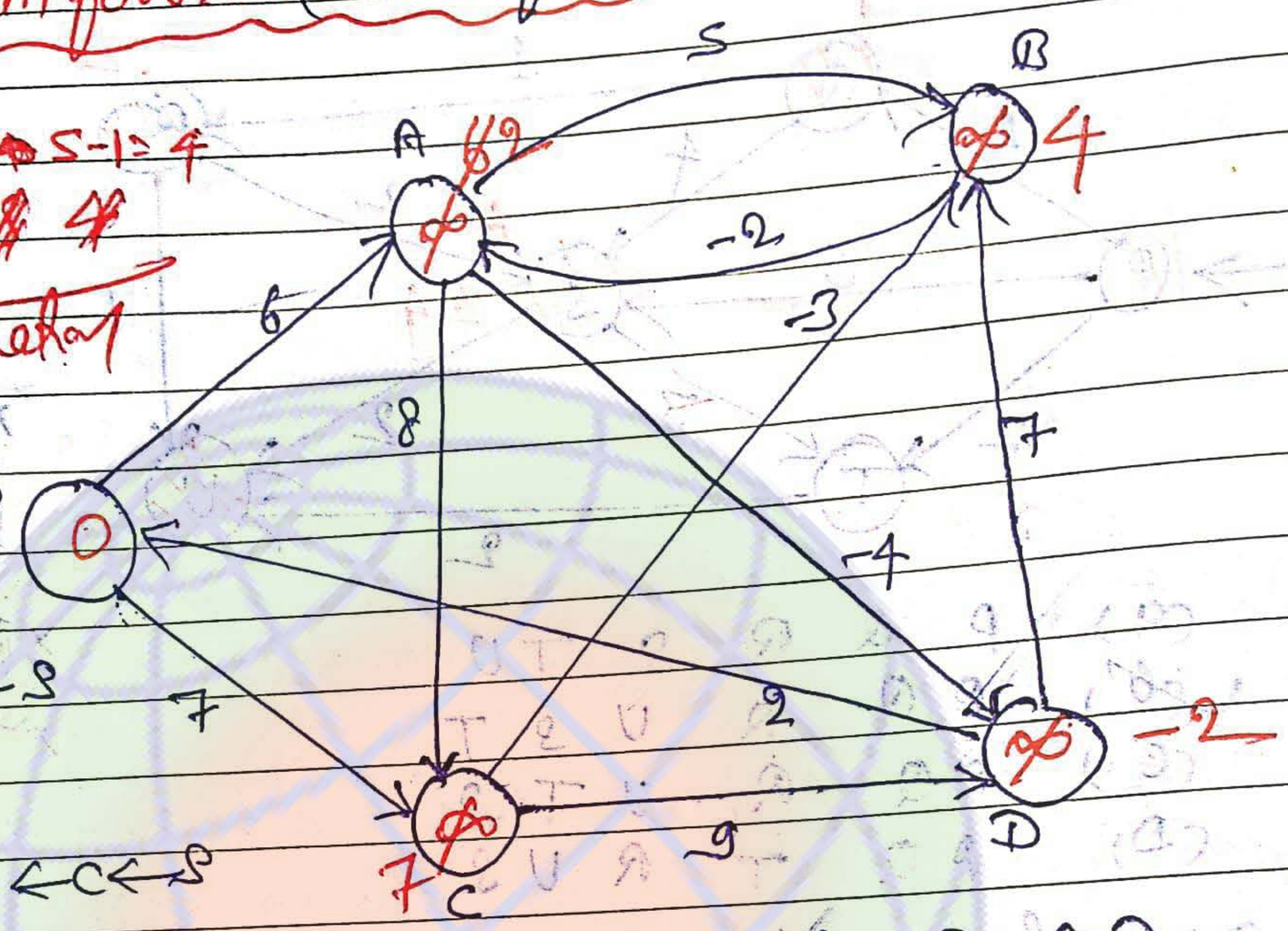
Date __/__/____ Page No.: ____

★ Ballman ford :- (Ballman-ford) :-

n-1 = 5-1 = 4

Q.) ~~4~~ ~~4~~ ~~4~~ 4

4 iteration



Path :-

A ← B ← C ← S

B ← C ← S

C ← S

D ← A ← B ← C ← S

v	S	A	B	C	D
π[v]	—	B	C	S	A
d[v]	0	2	4	7	-2

Bellman ford (G, w, s)

{

Initialize-Single source (G, w, s)

for i ← 1 to |V(G)| - 1

for each edge (u, v) ∈ E(G)

relax (u, v, w)

for each edge (u, v) ∈ E(G)

if [d[v] > d[u] + w(u, v)]

return false

return true

}

Date __/__/____ Page No.: ____

(i) Ballman ford remove the disadvantage of dijkastra and can work on -ve weights.

(ii) -ve weight cycle:-
A cycle is said to be -ve weights if the net/total weight of cycle is -ve or less than zero

(iii) If a graph have -ve weight cycle then ballman ford guarantee detection.

(iv) n-1 iteration are sufficient for a graph which have n-vertices because the longest path length is n-1.

(v) The -ve weight cycle will be detected but it must be rechable.

(vii) Even if all the weight of edge is distinct still there may be different path possible.

(viii) If Cost of all edge weights get doubles then the value of path may change but shortest path will remain same.

Date ___/___/______

Short trick to solve floyd warshell problem.

(i) self loop को Remove कर दें

(ii) अगर दो vertex के बीच दो Path हो तो smallest को select करें

(iii) D_0 → Normal distance को पता करें, तथा T_0 में जो भी Initial vertex हो उसे लिखें।

(Diagonal वाले में Zero ही होगा)

(iv) D_1 → या तो directly vertex 1 को intermediate मान कर find करें या फिर (or) → D_1 → 1st matrix/ D_2 → 2nd

— R0, C0 ले कर उनसे से ∞ वाले number को छोड़ दें।

→ अगर इसमें भी ∞ आ जाए तो उस row या column का सारा element as it is होगा क्योंकि, ∞ में कुछ भी direct जोड़े उससे कोई फर्क नहीं पड़ेगा

like,

for D1 → 3-1-4 → distance from (3 to 4 via 1)

3-1-2 etc., short trick for direct

Chitra

"You must be the change you want to see in the world." —Mahatma Gandhi

Date 26/06/2015 Page No.:

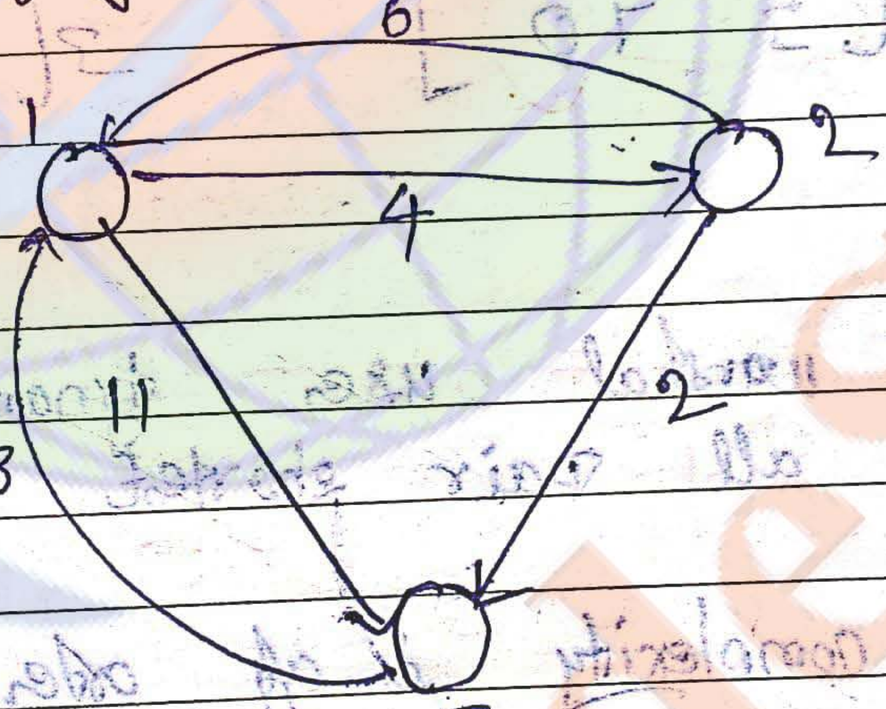
★ All pair shortest path:-

Let G be a directed weighted graph and the problem is to find the shortest distance from every node, to every other node of the graph.

(1) Either this problem can be solve using single source shorted path algorithm, if executed on every vertex. otherwise we can ~~first~~ floyd warshell algo, which uses dynamic Programming strategies to solve this problem.

(1) Floyd warshell algorithm:-

Q:> Consider a graph and find all pair shortest path using floyd warshal algorithm



$$D^0 = \begin{matrix} & 1 & 2 & 3 \\ 1 & 0 & 4 & 11 \\ 2 & 6 & 0 & 2 \\ 3 & \infty & \infty & 0 \end{matrix}$$

$$\Pi^0 = \begin{matrix} & 1 & 2 & 3 \\ 1 & - & 1 & 1 \\ 2 & 2 & - & 2 \\ 3 & - & - & - \end{matrix}$$

Date / /

d_1 =

	1	2	3
1	0	4	11
2	6	0	2
3	3	7	0

when 1 is intermediate

π =

	1	2	3
1	-	1	1
2	2	-	2
3	3	1	-

d^2 =

	1	2	3
1	0	4	6
2	6	0	2
3	3	7	0

when 2 is intermediate

π^2 =

	1	2	3
1	-	1	2
2	2	-	2
3	3	1	-

d^3 =

	1	2	3
1	0	4	6
2	5	0	2
3	3	7	0

when 3 is intermediate (This is final answer)

π =

	1	2	3
1	-	1	2
2	3	-	2
3	3	1	-

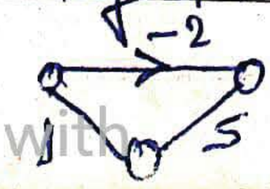
(i) Floyd warshal uses dynamic Programming to solve all pair shortest Path.

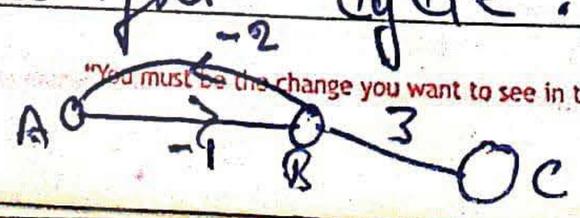
(ii) Its complexity is of order n^3, $O(n^3)$ and can work only with positive (+ve) weight edge.

(iii) If we have a graph of n-vertices then we require $2(n+1)$ matrix each of order $n \times n$.

Note

Supports negative-edge-weight, but not negative-weight cycle.





Chitra

Q-
A-
L-

R-
V-
T-

Date __/__/__ Page No.: _____

* Longest common sub-sequence:-

(i) In LCS (longest common sub-sequence) there are two string's x and y, and the problem is to find the longest common sub-sequence among them.

(ii) It's application is in DNA Pattern matching to find the similarity b/w two sells.

(iii) Step 1:-

Define the optimal structure

Let us consider two string x and y.

$x = x_1\ x_2\ x_3 \dots x_m$

$y = y_1\ y_2\ y_3 \dots y_n$

and the soln - b/w

$Z_1 \dots Z_q$

(iv) Case 1st:-

If $x_m = y_n$ then $x_m = y_m = Z_q$ ✓

LCS (x_m, y_n)

else If $(x_m \neq y_m)$

$x_m = y_n = Z_q$

LCS (x_m, y_m)

Date __/__/____ Page No.: ____

Case 2nd :-

If $x_m \neq y_n$, $x_m \neq z_q$

$\rightarrow$ LCS (x_m, y_n)

Case 3rd :-

If $(x_m \neq y_n)$ $y_n \neq z_q$

$\rightarrow$ L.C.S (x_m, y_{n-1})

e.g Consider two strings x and y.

$x = A\ B\ C\ B\ D\ A\ B$

$y = B\ D\ C\ A\ B\ A$

y		0	1	2	3	4	5	6
			B	D	C	A	B	A
x	0	0	0	0	0	0	0	0
1	A	0	0	0	0	1↑	1←	1↑
2	B	0	1↑	1←	1←	1←	2↑	2←
3	C	0	1↑	1↑	2↑	2←	2↑	2↑
4	B	0	1↑	1↑	2↑	2↑	3↑	3↑
5	D	0	1↑	2↑	2↑	2↑	3↑	3↑
6	A	0	1↑	2↑	2↑	3↑	3↑	4↑
7	B	0	1↑	2↑	2↑	3↑	4↑	4↑

B C B A

Date __/__/____ Page No.: _____

Step 2nd: This is recursive soln for L.C.S

$$c[i,j] = \begin{cases} 0 & i=0, \text{ or } j=0 \\ c[i-1, j-1]+1 & i,j>0, x_i = y_j \\ \max[c[i,j-1], c[i-1,j]] & i,j>0, x_i \neq y_j \end{cases}$$

Step 3:

```
LCS (X, Y)
    m ← length[X]
    n ← length[Y]
    for i ← 0 to m
        c[i,0] ← 0
    for j ← 0 to n
        c[0,j] ← 0
    for i ← 1 to m
        for j ← 1 to n
            if (xi == yj)
                c[i,j] ← c[i-1,j-1] + 1
                b[i,j] ← "↖"
            else if c[i-1,j] ≥ c[i,j-1]
                c[i,j] ← c[i-1,j]
                b[i,j] ← "↑"
            else
```

Date __/__/____ Page No.: ____

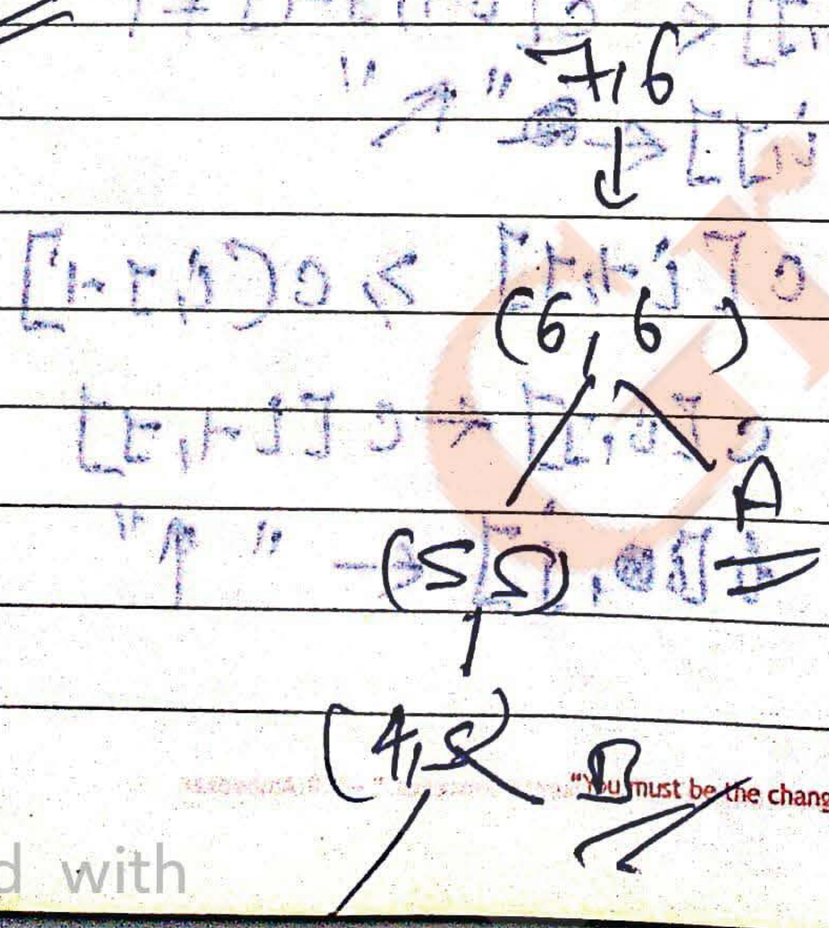
c[i,j] ← c[i,j-1]

b[i,j] ← "←"

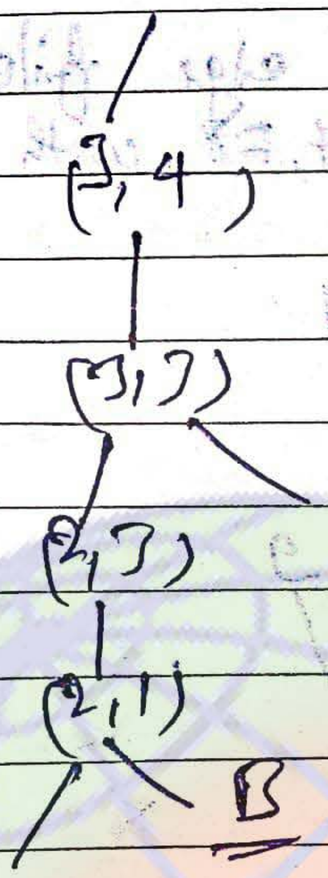
Step 4:

Construct the final solution:-

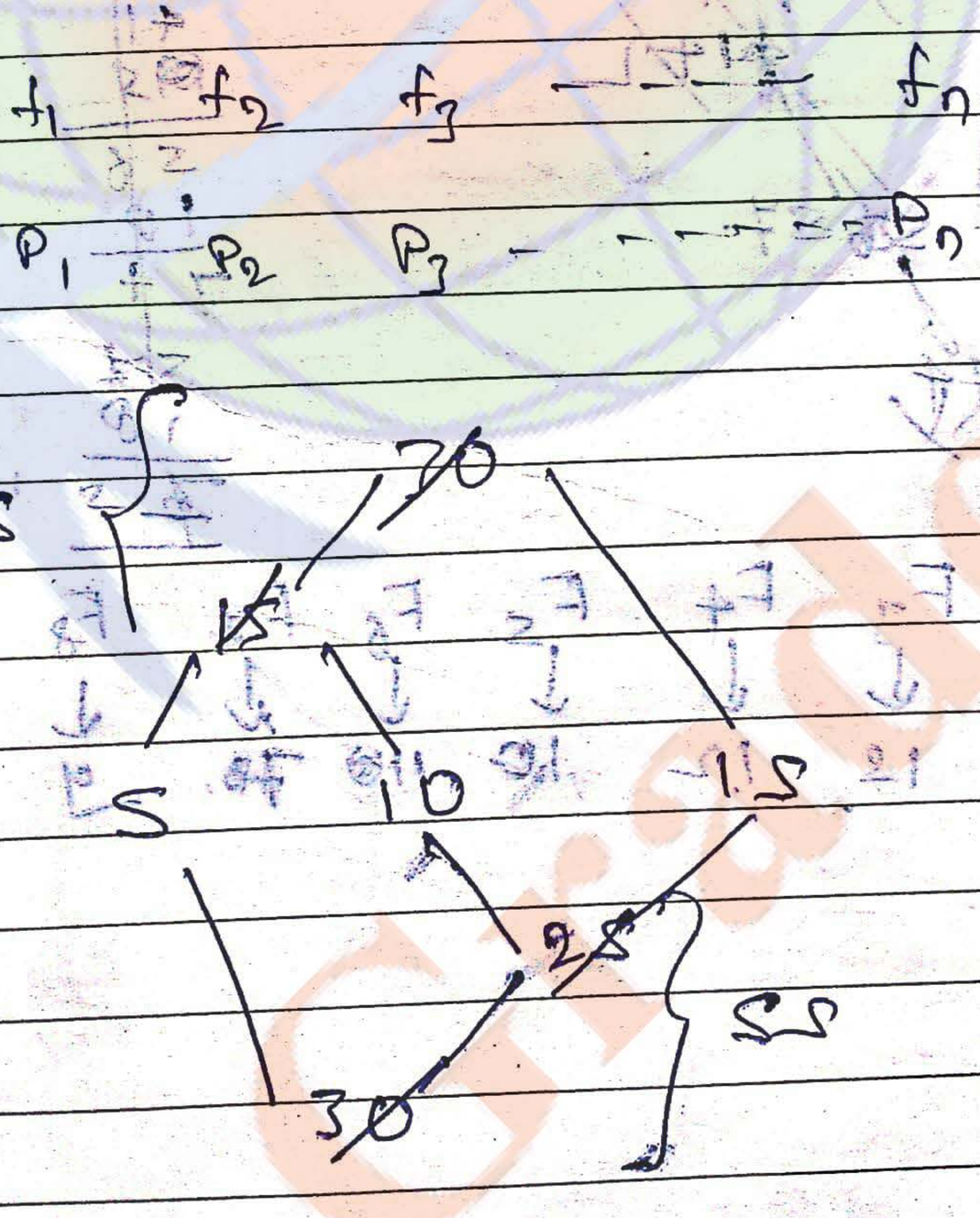
```
Print_LCS (b, X, i, j)
  if (i==0 || j==0)
     return
  if [b[i,j] == "↖"]
     print Xi
     LCS [b, X, i-1, j-1]
  else if [b[i,j] == "↑"]
     LCS [b, X, i-1, j]
  else
     LCS [b, X, i, j-1]
```



Date __/__/____ Page No.: _____



☆ Optimal merge Pattern :-



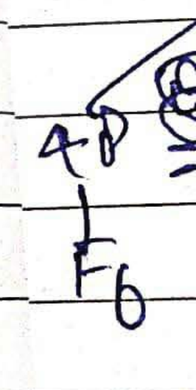
Date __/__/____ Page No.: ____

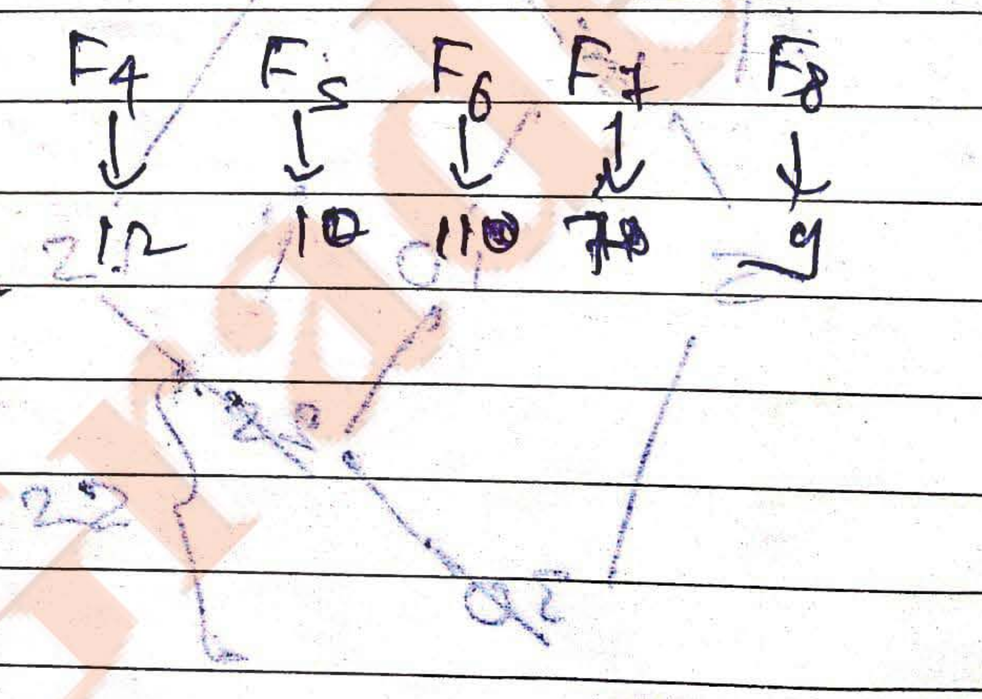
Q) Consider the following edge files
F1, F2, F3, F4, F5, F6, F7, F8 with following pages
18, 3, 15, 12, 10, 11, 7, 9

Sol^n

Sol^n

F1	F2	F3	F4	F5	F6	F7	F8
18	3	15	12	10	11	7	9

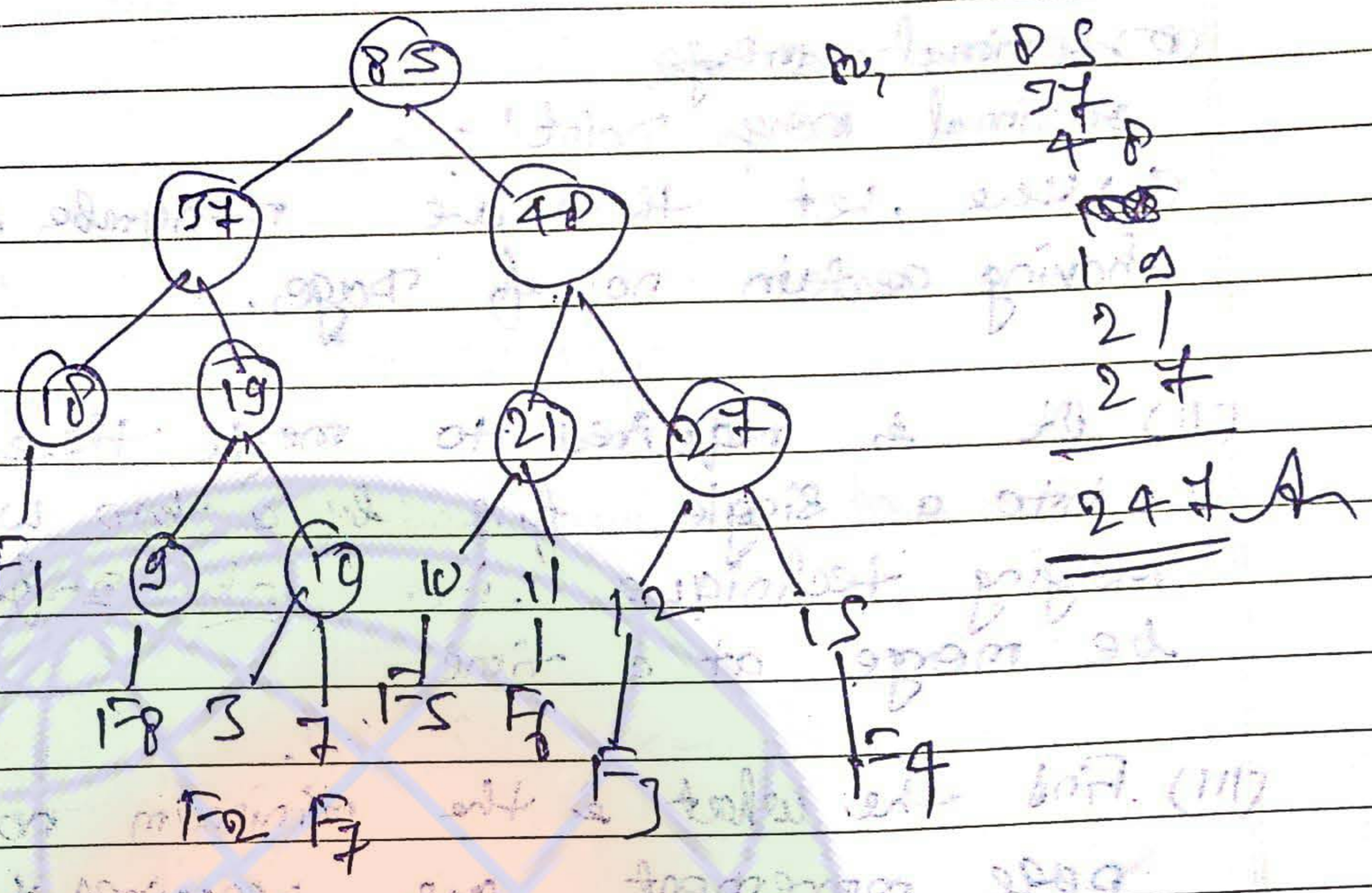




Chitra

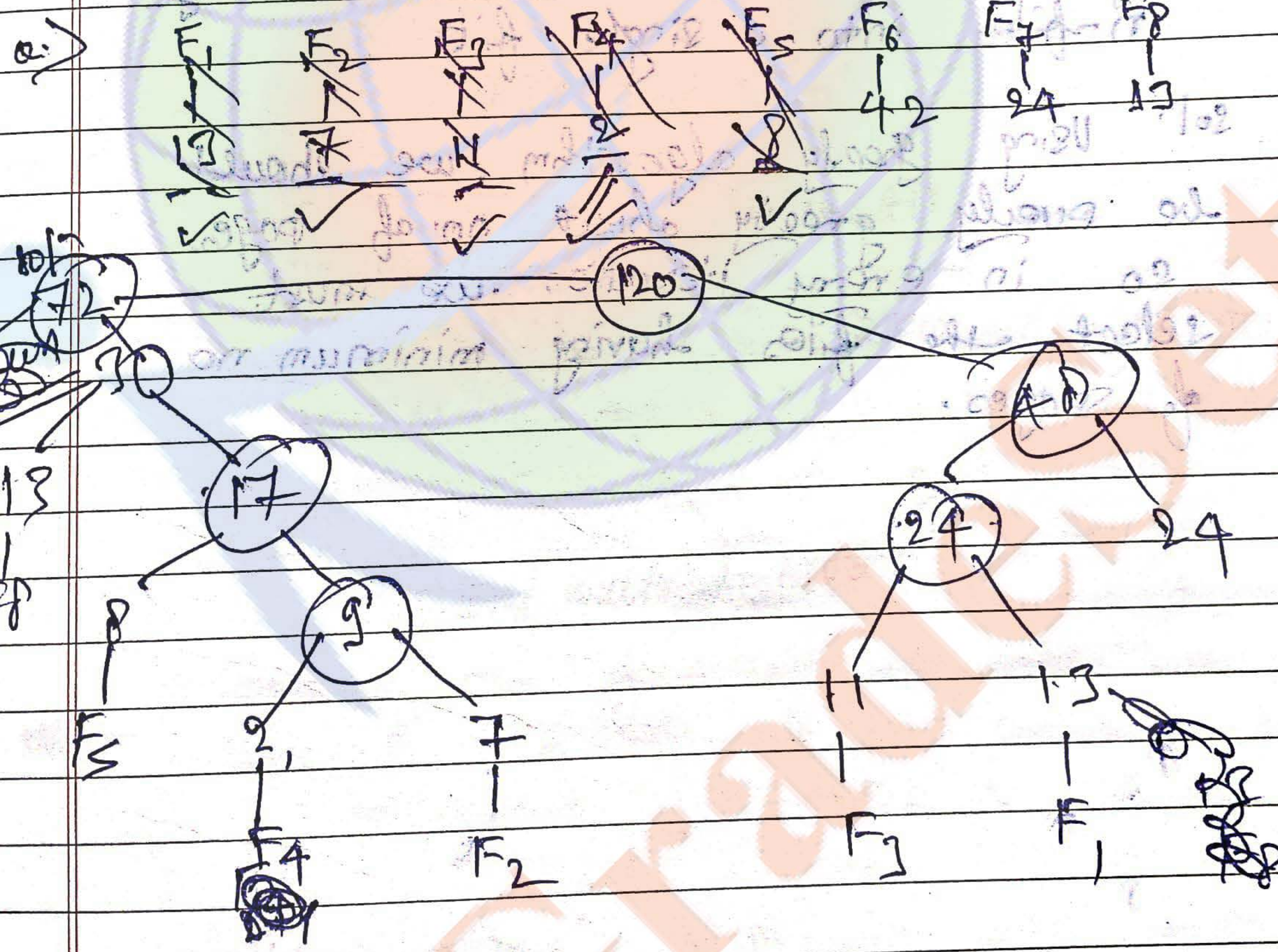
"You must be the change you want to see in the world." —Mahatma Gandhi

Date __/__/____ Page No.: ____



DS
77
40
21
27
24 7 A

Q.)



"Cultivation of mind should be the ultimate aim of human existence." —B.R.Ambedkar

Date __/__/____ Page No.: ____

~~Optimal merge~~

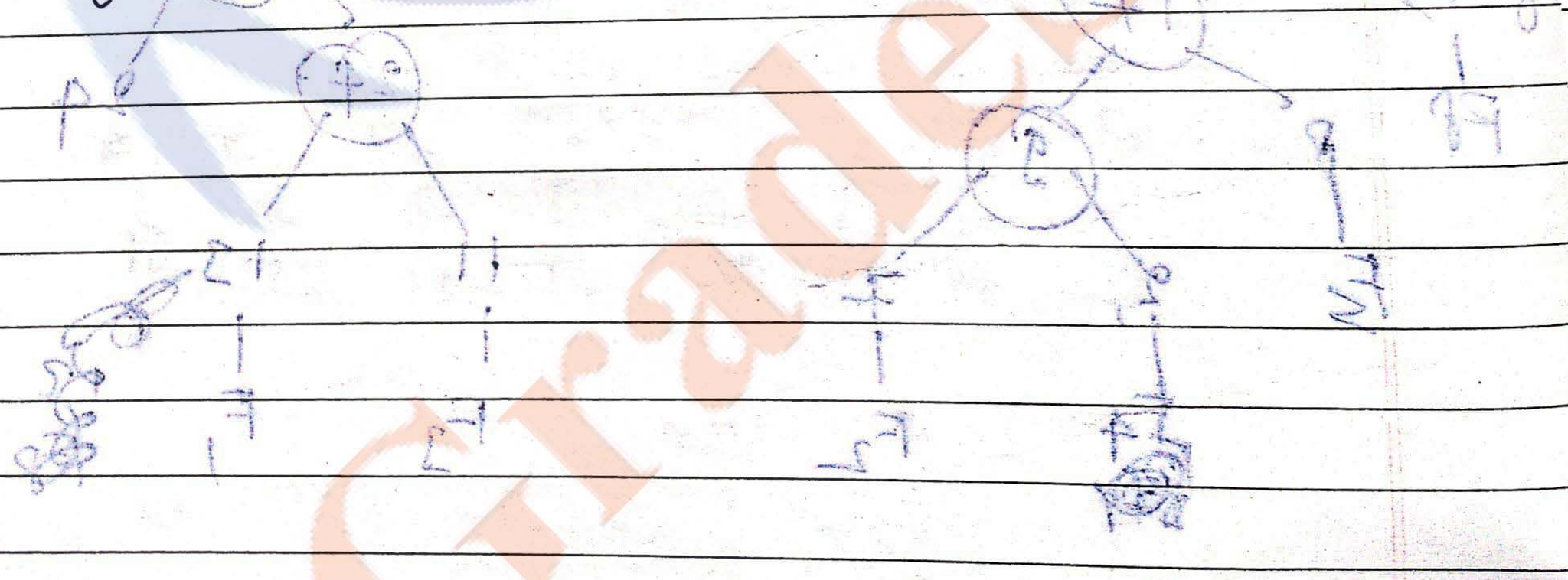
Optimal merge Point! -

(i) Here, let there are n-numbe of files having certain no. of pages.

(ii) It is requires to merge these files into a single file by a two way merging technique, i.e. only 2-files can be merge at a time.

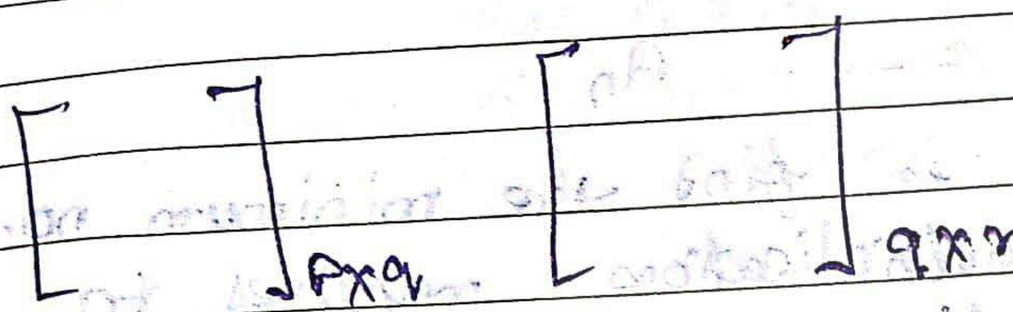
(iii) Find the what is the minimum no. of page movement are requires to merge n-files into a single file

Soln Using greedy algorithm, we shows be purely greedy about no. of pages. so in every iteration, we must select the files having minimum no of pages.

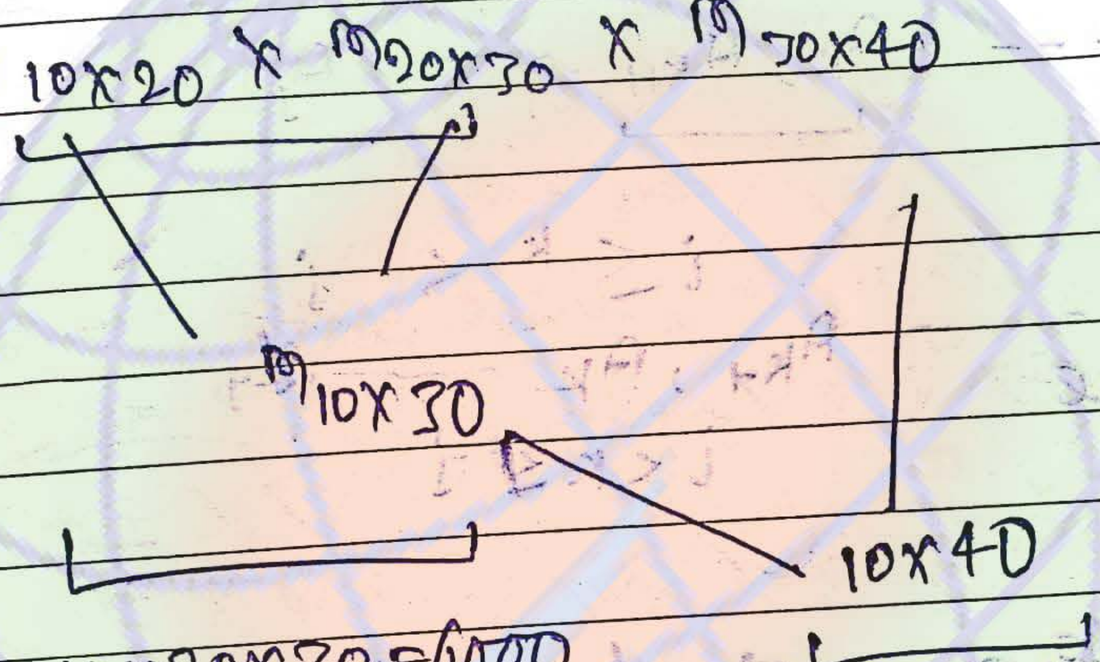


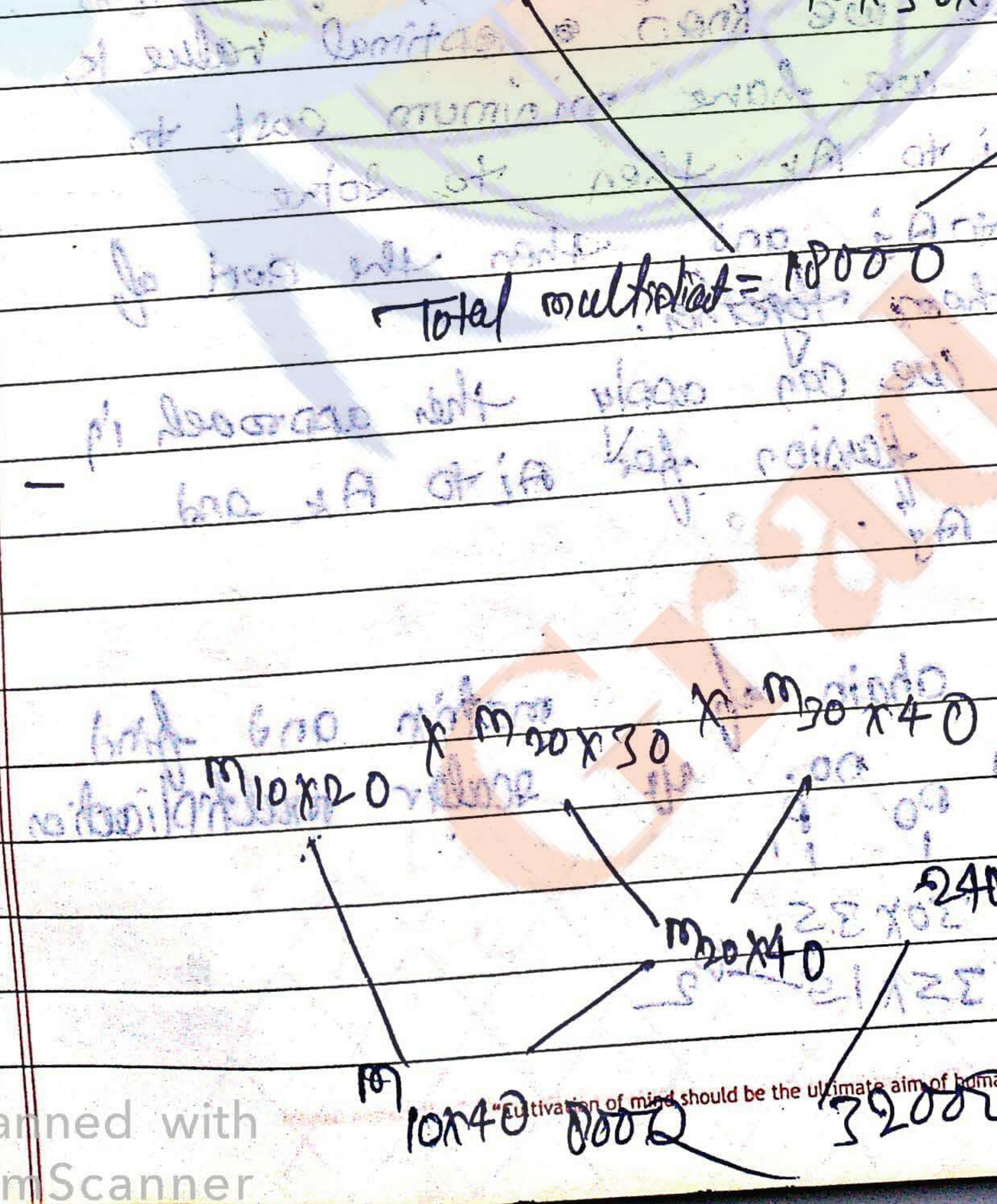
Date __/__/____ Page No.: ____

★ matrix Chain multiplication :-

1) 

Total No of scalar multiplication = PXqXr

eg — 

eg — 

Date __/__/____ Page No.: ____

Q) Consider a chain of matrix from

$A_1, A_2 \dots A_n$

the problem is find the minimum no. of scalar multiplications requires to solve the problem. also find the exact order in which matrix will be multiplied.

$A_i \dots A_k, A_{k+1} \dots A_j$

$i \le k < j$

$A_{k+1}, A_k \dots A_q$

$i < k \le j$

- optimal structure

Step 1:-

Let there be a chain of matrix from A_i to A_j, suppose we know a optimal value k, such that we have minimum cost to solve A_i to A_k then to solve A_{k+1} to A_j and then the cost of multiplying them together.

And then we can apply this approach in a recursive fasion for A_i to A_k and for A_{k+1} to A_j.

Q) Consider a chain of matrix and find the minimum no. of scalar multiplication requires.

	P_0	P_1
A_1	30	35
A_2	35	15

P_2

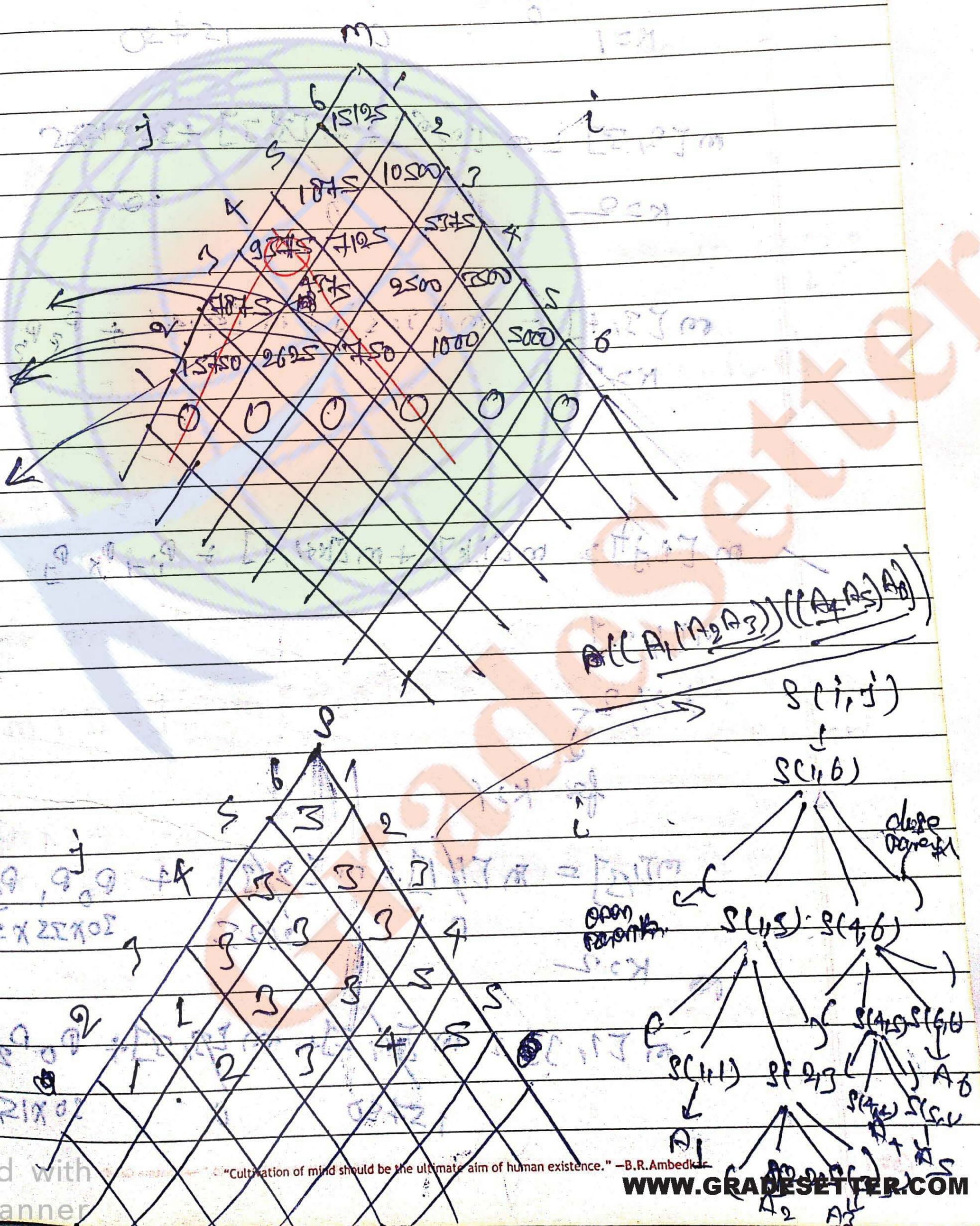
Date __/__/____ Page No.: ____

A_3 15 x 5 — P_3

A_4 5 x 10 — P_4

A_5 10 x 20 — P_5

A_6 20 x 25 — P_6



$-m[1,1]=0$

$-m[1,2]=$

$1 \le k < 2$

$P((A_1(A_2A_3))((A_4A_5)A_6))$

$S(i,j)$

$S(1,6)$

Date __/__/____

$$m[i,j] = m[i,k] + m[k+1,j] + p_{i-1} p_k p_j$$

$$m[1,2] = m[1,1] + m[2,2] + p_0 p_1 p_2 = 0 + 0 + 10 \times 15 \times 20 = 3000$$

$$m[2,3] = m[2,2] + m[3,3] + p_1 p_2 p_3 = 0 + 0 + 15 \times 20 \times 25 = 7500$$

$$m[3,4] = m[3,3] + m[4,4] + p_2 p_3 p_4 = 0 + 0 + 20 \times 25 \times 30 = 15000$$

$$m[1,3] = \min \begin{cases} m[1,1] + m[2,3] + p_0 p_1 p_3 \\ m[1,2] + m[3,3] + p_0 p_2 p_3 \end{cases}$$

$$= \min \begin{cases} 0 + 7500 + 10 \times 15 \times 25 \\ 3000 + 0 + 10 \times 20 \times 25 \end{cases}$$

$$= \min \begin{cases} 11250 \\ 8000 \end{cases} = 8000$$

$$m[2,4] = \min \begin{cases} m[2,2] + m[3,4] + p_1 p_2 p_4 \\ m[2,3] + m[4,4] + p_1 p_3 p_4 \end{cases}$$

$$= \min \begin{cases} 0 + 15000 + 15 \times 20 \times 30 \\ 7500 + 0 + 15 \times 25 \times 30 \end{cases}$$

$$= \min \begin{cases} 24000 \\ 18750 \end{cases} = 18750$$

$$m[1,4] = \min \begin{cases} m[1,1] + m[2,4] + p_0 p_1 p_4 \\ m[1,2] + m[3,4] + p_0 p_2 p_4 \\ m[1,3] + m[4,4] + p_0 p_3 p_4 \end{cases}$$

$$= \min \begin{cases} 0 + 18750 + 10 \times 15 \times 30 \\ 3000 + 15000 + 10 \times 20 \times 30 \\ 8000 + 0 + 10 \times 25 \times 30 \end{cases}$$

$$= \min \begin{cases} 23250 \\ 24000 \\ 15500 \end{cases} = 15500$$

Date / /

$m[i,j] = m[i,k] + m[k+1,j] + P_{i-1} P_k P_j$

$i \le k < j$

$m[2,4]$

$k = 2, 3$

for $k=2$

$m[2,4] = m[2,2] + m[3,4] + P_1 P_2 P_4$

0 750 $= 15 \times 15 \times 10 = 6000$

for $k=3$

$m[2,4] = m[2,3] + m[4,4] + P_1 P_3 P_4$

2625 0 $35 \times 5 \times 10 = 4375$

$m[i,j] = m[i,k] + m[k+1,j] + P_{i-1} P_k P_j$

$i \le k < j$

$m[3,5]$

$k = 3, 4$

for $k=3$

$m[3,5] = m[3,3] + m[4,5] + P_2 P_3 P_5$

$= 2500$

for $k=4$

$m[3,5] = m[3,4] + m[5,5] + P_2 P_4 P_5$

Page No.:

Date

$m[i,j] = m[i,k] + m[k+1,j] + P_{i-1} P_k P_j$

$i \le k < j$

$m[1,4] =$

$1 \le k < 3$

for $k=1$

$m[1,4] = m[1,1] + m[2,4] + P_0 P_1 P_4$

for $k=2$

$m[1,$

$+ P_0 P_2 P_4$

for $k=3$

$+ P_0 P_3 P_4$

$+ P_0 P_1 P_4$

$= 9375$ ✓

$m[i,j] = m[i,k] + m[k+1,j] + P_{i-1} P_k P_j$

$i \le k < j$

$m[1,5]$

$k = 1, 2, 3, 4$

for $k=1$

$m[1,5] = m[1,1] + m[2,5]$ —

Date __/__/____ Page No.: _____

m [1, 6]

K = 1, 2, 3, 4, 5

for K = 3

m [1, 3] + m [4, 6] + P0 P3 P6

7842 + 3500

30 x 85 x 25 = 63750

$$m[i,j] = \begin{cases} 0 & i == j \\ \min\{m[i,k] + m[k+1,j] + P_{i-1} P_k P_j\} & i < j \\ i \le k < j \end{cases}$$

Print Parenthesis(i, j)

{

if (i == j)

{

print Ai

print ")"

print "("

Date __/__/____ Page No.: ____

Print_Parenthesis (i, j, s[i,j])

Print ~ " " (s[i,j]+1, j)

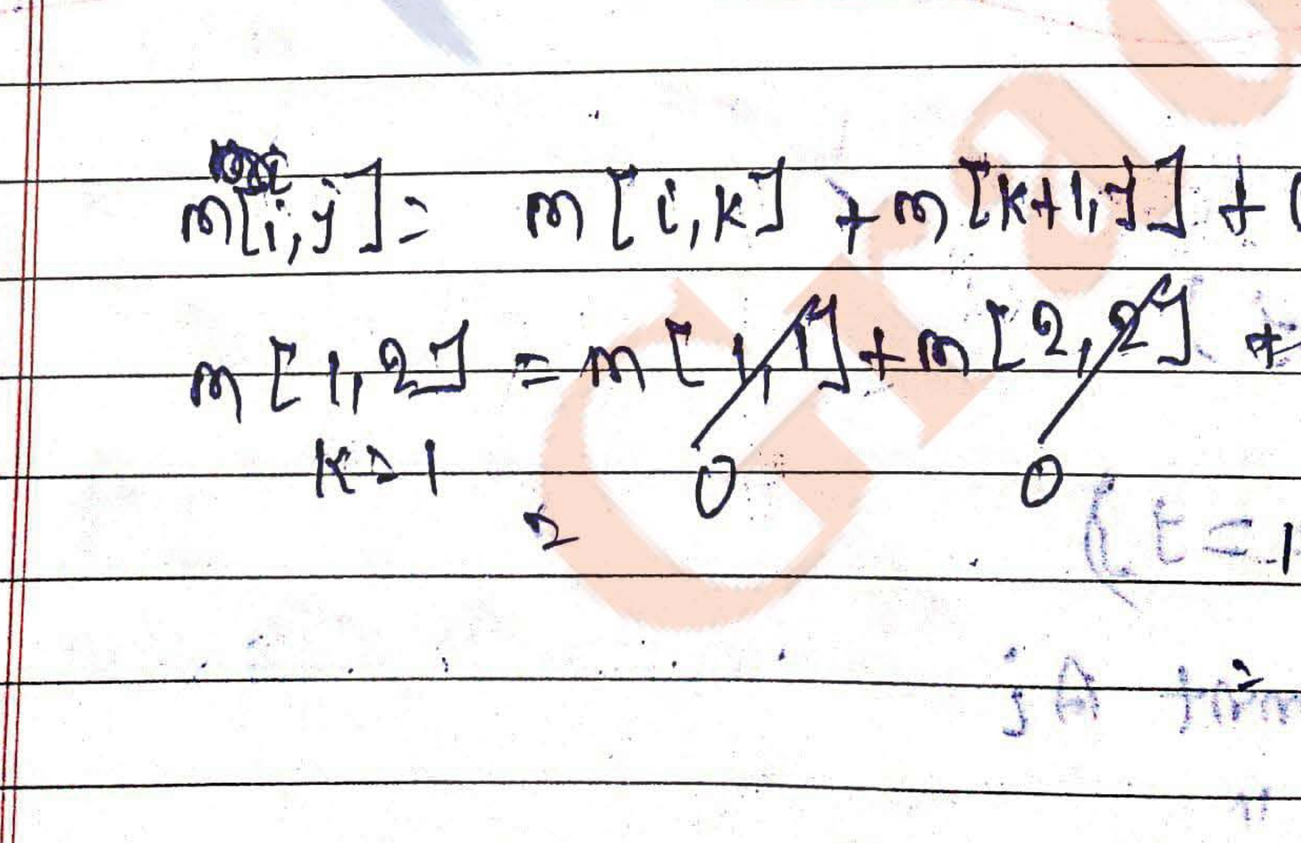
Q) m_1 10×100 p_0 p_1 (a) 2,40,000

m_2 100×20 p_1 p_2 (b) 44000

m_3 20×5 (c) 19000

m_4 5×80 (d) 25000

soln



$P_0 = 10$

$P_1 = 100$

$P_2 = 20$

$P_3 = 5$

$P_4 = 80$

$m[i,j] = m[i,k] + m[k+1,j] + P_{i-1} P_k P_j$

$m[1,2] = m[1,1] + m[2,2] + P_0 P_1 P_2$

$k=1$

$= 0 + 0 + 10 \times 100 \times 20$

$= 20000$

Date / /

Page No.:

$m[2,3] = m[i,k] + p\,m[k+1,j] + p_{i-1} p_k p_j$

$k=2$

$= m[2,2] + m[3,3] + p_1 p_2 p_3$

$= 0 + 0 + 100 \times 20 \times 5$

$= 10,000$

$m[$

10000

$20,000$

$30,000/2 = 15000$

$m[1,4] = m[1,1] + m[4,4] + p_0 p_1 p_4$

$= 15000 + 0 + 10 \times 5 \times 80$

$= 15000 + 4000$

$= 19000$

Q.) Consider a string $A = qpqrs$

$B = pqprsq$

Let x be the length of longest common subsequence and y are the pair such common subsequence find the value of $x+10y$.

Soln

Page No.:

Date __/__/____

Soln: length of longest common subsequence of p & q

p q

{ q p q r
  p q r r
  q p r r }

4 + 10 x 3 = 34 Ans

⇒ An algorithm to find

A [0 ——— n-1]

i

[0 1 ≤ 2 8 12 14 17 19]

$l_{n-1} = 1$

∀ i ≤ n-2

$$L_i = \begin{cases} 1 + L_{i+1} & \text{if } A[i] < A[i+1] \\ 0 & \text{otherwise} \end{cases}$$

Get max [L_0, L_1 —— L_{n-1}]

↳ Dynamic

Date __/__/____ Page No.: ____

Sum of Subset Problem :- $\{2, 1, 4 \ldots\}$

$m - 2^m$

$\log_2 m \quad m$

★ Huffman Coding :-

Consider a coding system, which have 5 symbols $m_1, m_2 \ldots$ with Probability

$m_1 - 0.12$

$m_2 - 0.04$

$m_3 - 0.45$

$m_4 - 0.16$

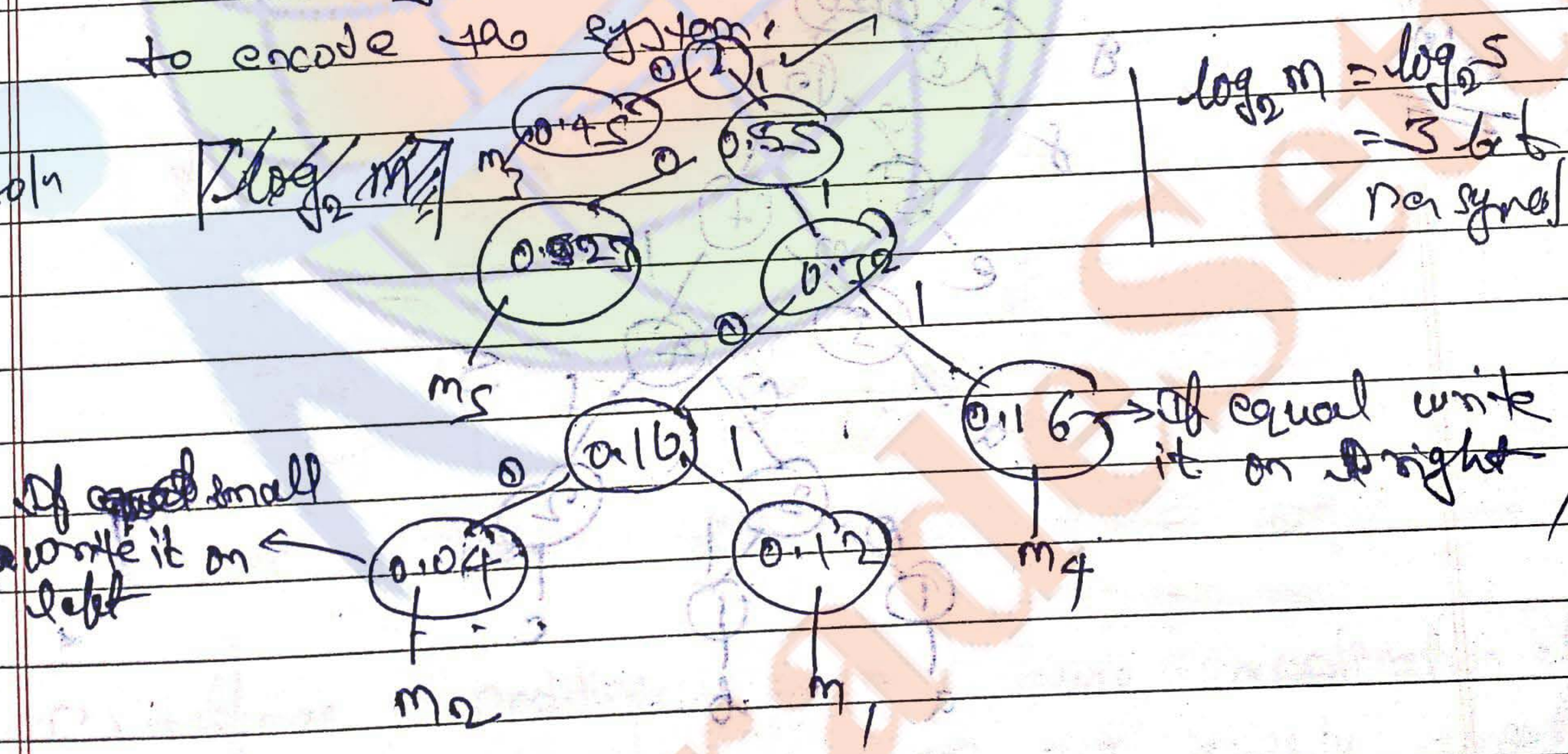
$m_5 - 0.23$

How many bits per symbol are required to encode the system?

soln $\lceil \log_2 m \rceil$

$\log_2 m = \log_2 5$

$= 3$ bits per symbol



$m_1 - 110$ $\quad 2 \times 1 = 0.12 \times 4 = 0.48$

$m_2 - 1100$ $\quad 2 \times 1 = 0.04 \times 4 = 0.16$

$m_3 - 00$ $\quad 2 \times 2 = 0.45 \times 1 = 0.45$

$m_4 - 111$ $\quad = 0.16 \times 3 = 0.48$

$m_5 - 10$ $\quad = 0.16 \times 2 = 0.32$

$2.05 = 0.64$

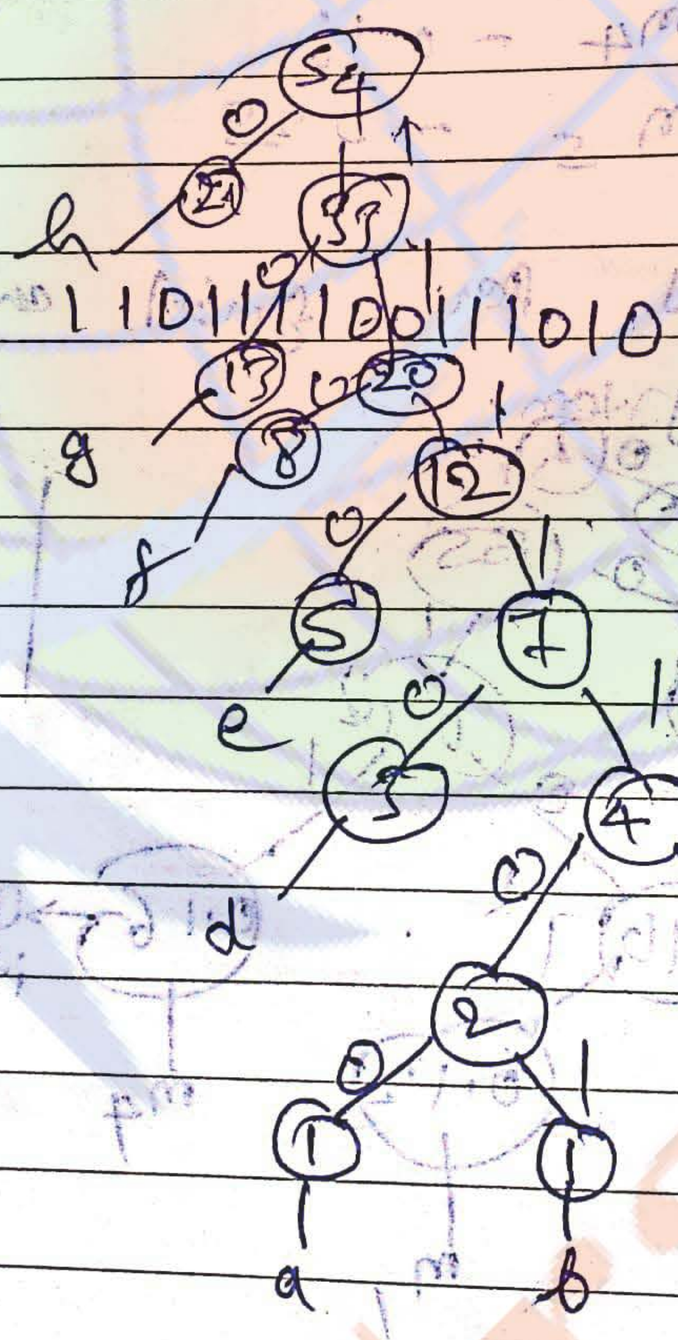
Date __/__/____ Page No.: ____

Conclusion:-

(i) With Huffman Coding 2.03 bits per symbol is requires

(ii) Total reduction is 0.97 bits per symbol.

Q> Consider 8 characters from a-1, b-1, c-2, d-3, e-5, f-8, g-13, h-21



a = 1111 = 1100 = 1 x 5 = 5 bits

b = 1101 = 1100 = 1 x 5 = 5 bits

e = 1111 = 10 = 2 x 6 = 12 bits

d = 1110

c = 1110

Date __/__/____ Page No.: ____

f = 110 →
g = 10
h = 0

(1) Huffman coding is a pure application of greedy algorithm, where we are greedy about probability

(11) The no having the maximum probability should have the minimum no. of bits assign

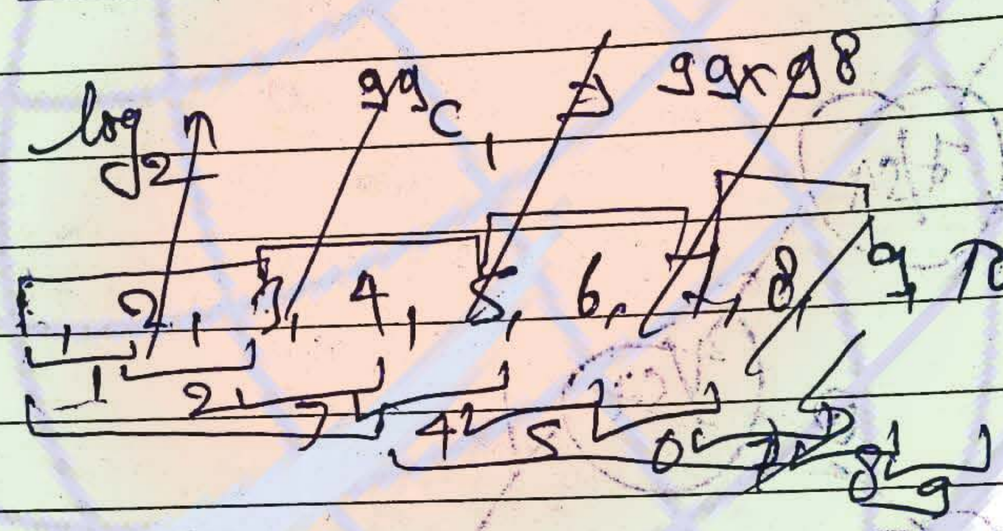
Date __/__/____

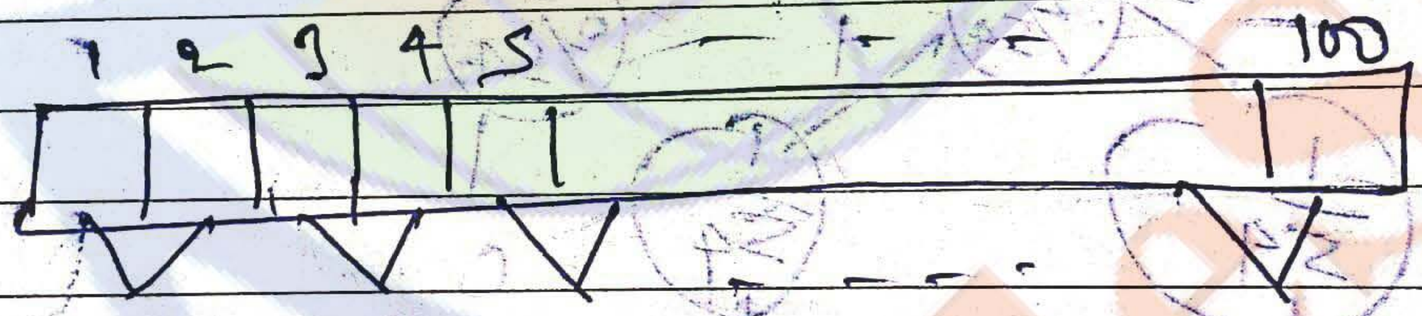
Page No.: ____

(iii) Ot guarantee to give minimum bits per symbol bases on Probability.

2014
Q> The minimum no of comparision requires to find the min. and the max of 100 numbers.

Sol^n





At Compare
2 ans
then compare
there,

50	already compare
25	25
12	12
6	6
3	3
2	2
1	1
99	49

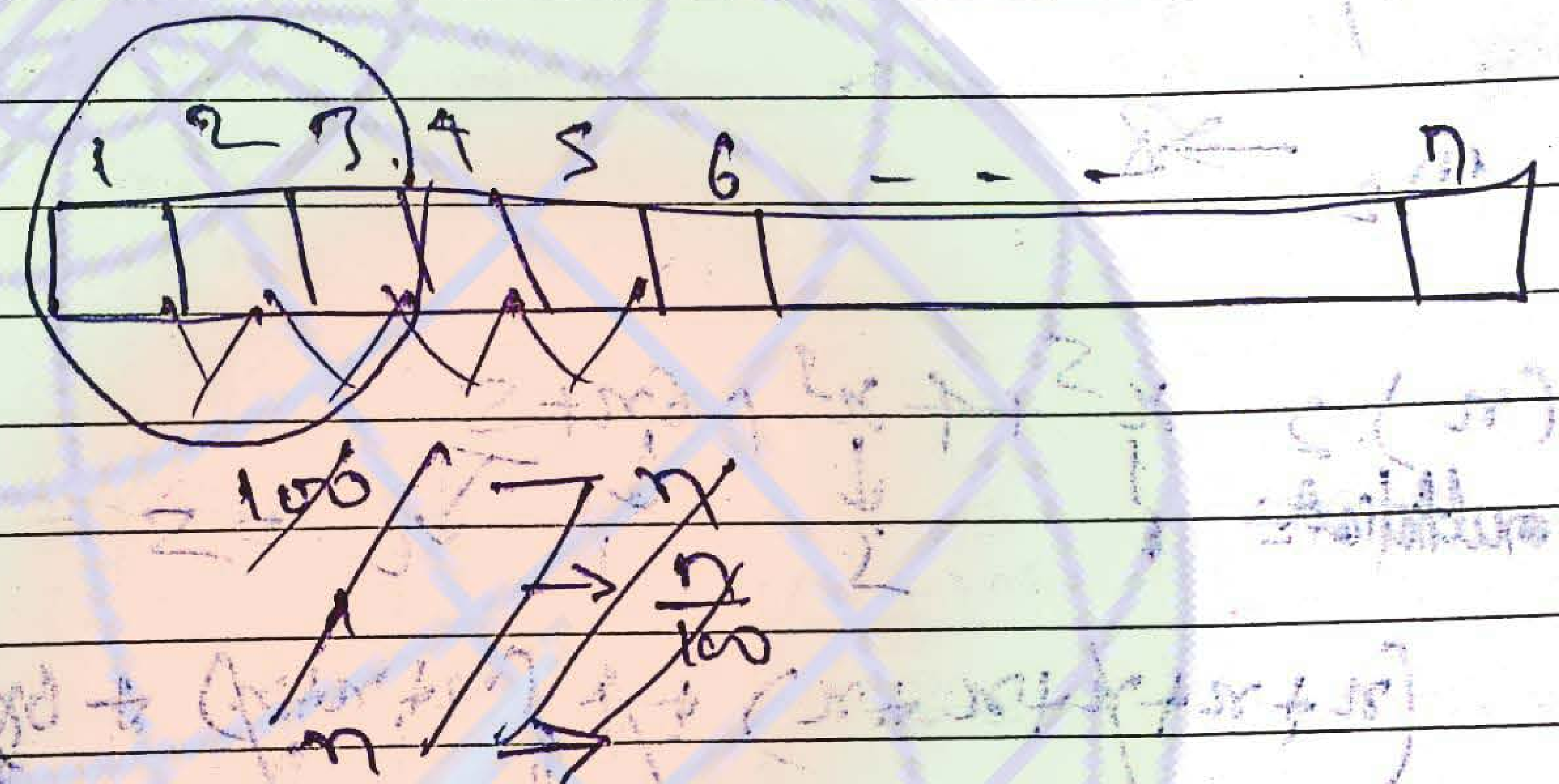
find maximum — min

⇒ 148 Ans

Date __/__/__ Page No.: ______

Q) Let unordered list contain n-distinct element.
What is the no. of comparision required to find a element which is neither minimum nor maximum.
(a) $O(n \log_2 n)$ (b) $O(n)$ (c) $O(\log_2 n)$
(d) $O(1)$

20/7



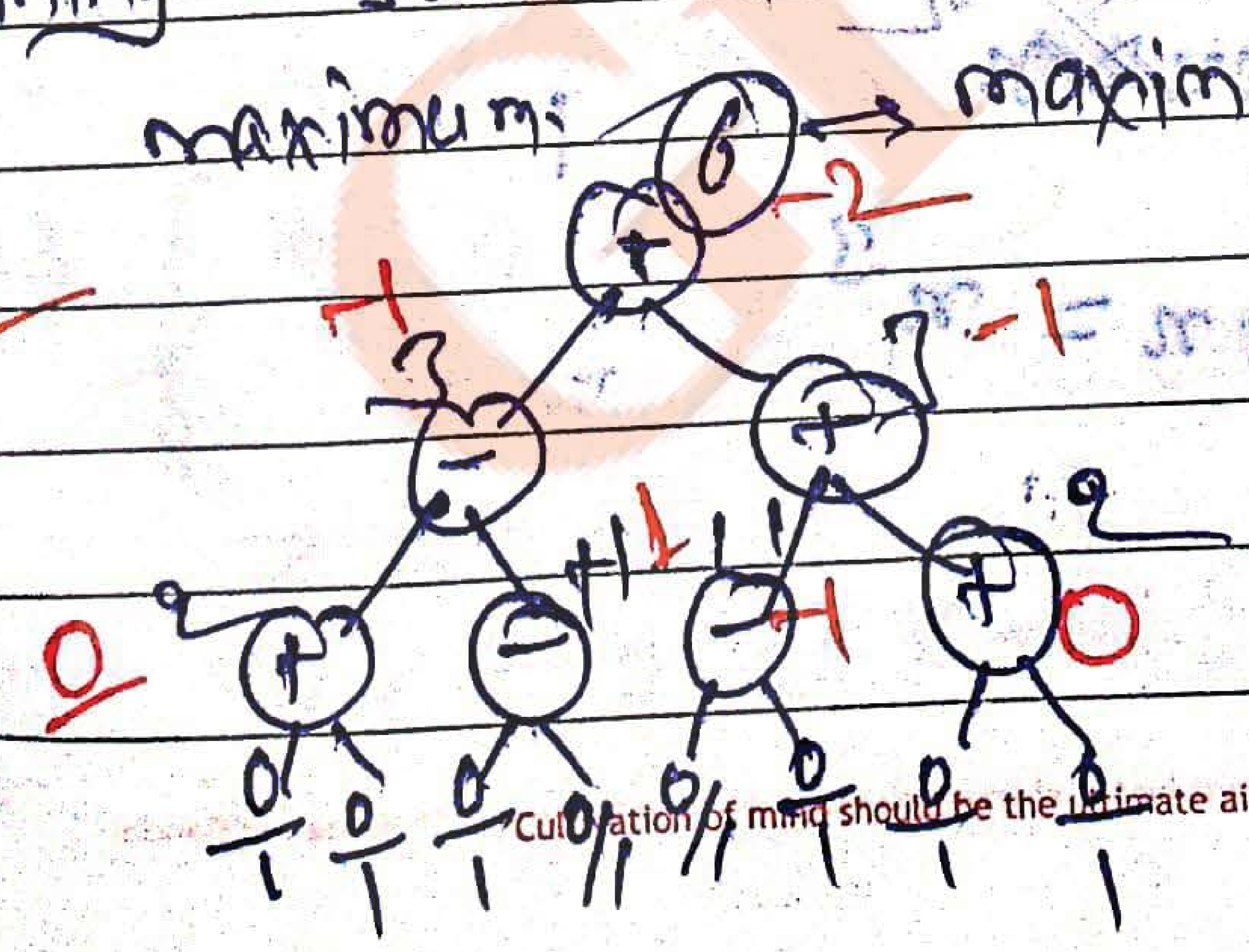
3 4 2 5 2 6

select 3 number in which one number is middle of 2,

In this case just compare any 3-value and find their min and maximum. The remaining 3rd number is for sure neither min nor maximum.

Q) max = 6
min = -2

min and maximum of



Date ___/___/_____ Page No.: _____

Q.) $V_{n \times m}$ $V[i,j] = i - j$

find the sum of all the elements in the array

(a) 0 (b) n-1 (c) $n^2 - 3n + 2$ (d) $n^2(n/2)$

Q) $m_1 \rightarrow A$

$m_2 \rightarrow B$

Q) $P(x) = x^5 + 4x^3 + 6x + 5$

multiplication = 1, 3, 1, 0 = 5

$(x + x + x + x + x) + A(x + x + x) + 6 \times x + 5$

1 2

$A = 2$

$\Rightarrow x^4(x^2 + 4x) + 6x + 5$

$\Rightarrow x^4\left[x^2(x^2 + 4)\right] + 6\Big] + 5$

$x \times x = x^2$

$x^2 \times x = x^3$

Q.>
```
bool fun(int n)
{
int sum=0
for (int odd=1; n>sum; odd+=2)
        sum = sum+odd
return (n==sum)
}
```

what is the return value of the function for 484.

Soln:

n =

1

1 = 1+1

(2)

2 = 2+2

= (4)

4 = 4+4

= 5

5 =

sum = 0 1 4 take n=4

odd = 1 3

0+1 = 1

1, 4, 9 —

Ans → Perfect square

2015 Q

Dot (x, a, n)

{

z ← 1

for i ← 0 to K-1

Date ___/___/_____ Page No.: _____

$z \leftarrow z^2 \bmod n$

if (e[i] == 1)
then
$z \leftarrow (z \times a) \bmod n$

}
return z

}

e = { e1 e2 — ek }

o/p

k

entries

if k = 4 , e = <10 11 7>

a = 2 n = 8

z ?

so 1

	0	1	2	3
e	1	0	1	1

$a^{11} \bmod 1 = 10$

z = 1

Ans = 0

Q) void fun(int arr[], int n)

{

for (int i=0; i<n; i=i+2)

{

if (i>0 && arr[i-1] > arr[i])

swap(&arr[i], arr[i-1])

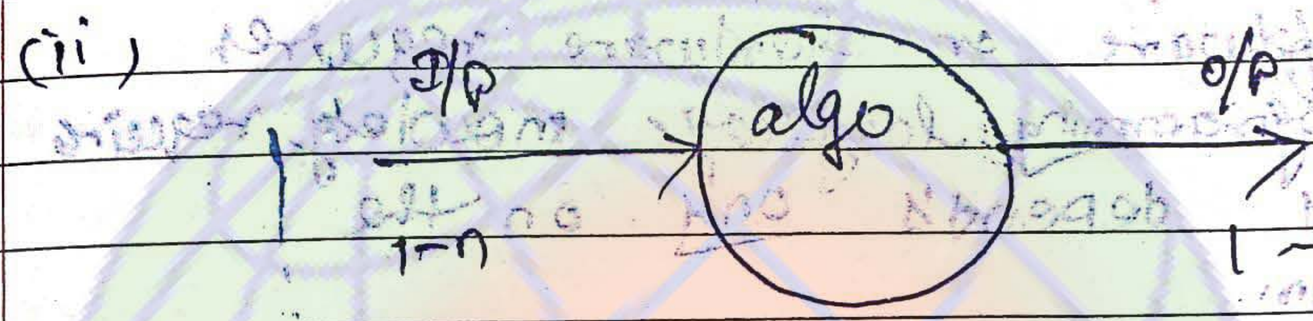
Date / / Page No.:

Analysis of algorithm

(i) Algorithm is a step by step procedure to solve a problem and here rough time prediction is also possible.

(The no. of steps will always be finite)

(ii)



The input to any algorithm will range from 0 to n and the output will range from 1 to m

(iii) Before an algorithm is converted into a program using a programming language it is necessary to do analysis of the algorithm

* Why we require analysis?

There are multiple resources which an algorithm requires like time, space, bandwidth, battery power, registers etc. out of which time is the most important criteria on which we will judge a algorithm.

Date __/__/____ Page No.: ____

* There can be two method for analysis of algorithm

(i) Independent or apriori analysis

Here we use special mathematical notation to give a rough time prediction.

Advantage:-
- No software or hardware requires
- No programming language knowledge requires
- Uniformly depends only on the algorithm.

(ii) Experimental analysis (or aposteriori analysis)

Here we convert a algo into a program using a programming language and then find exact time of execution.

Advantage:-
- Exact value, not rough prediction
- No mathematical knowledges requires.

Disadvantage:
Result is affected by many factor like hardware, software and programming language

Date __/__/____ Page No.: ____

* Order of magnitude:-

Order of magnitude of a instruction means how many fundamental ~~instruction~~ operation are requires to execute that instruction or statement

- Consider the following code and identify their order of magnitude/ rough time Prediction

(i) for i ← 1 to n; — $O(n)$
  S', — $O(1)$
$\Rightarrow O(n)$

(ii) for i ← 1 to n — $O(n)$
  for j ← 1 to n — $O(n)$
   S', — $O(1)$
nested → $O(n^2)$

(iii) for i ← 1 to n/2 — $O(n)$
  for j ← 1 to n/8 — $O(n)$
   S', — $O(1)$
nested → $O(n^2)$

(iv) for i ← 1 to n/2 — $O(n)$
  S', — $O(1)$ — $O(n)$
 for i ← 1 to n/16 — $O(n)$
  for i ← 1 to n — $O(n)$ — $O(n^2)$
→ $O(n^2)$

(v) j = 1
 while (j ≤ n)
 {
  j = j*2
 }
1, 2, 4, 8 — $O(\log_2 n)$

(v) for (i=1; i*i ≤ n; i++)
  S',
$i^2 \le n$, $i \le \sqrt{n}$ — 1, 4, 9, 16, 25 — $O(\sqrt{n})$
2×2 = 4
3×3 = 9

Chitra "Cultivation of mind should be the ultimate aim of human existence." —B.R. Ambedkar

Date __/__/____ Page No.: ____

(vi) for i <= 1 to n
  for (j=1; j<=n; j=j*2)

$O(n \log_2 n)$ ← $O(n)$, $\log_2 n$

(i) void fun(int n)
{
  int i, j, k, count=0
  for (i=n/2; i<=n; i++) — $O(n)$
    for (j=1; j+2<=n; j++) — $O(n)$
      for (k=1; k<=n; k=k*2) — $O(\log_2 n)$
  }

$O(n) \times O(n) \times O(\log_2 n) = O(n^2 \log_2 n)$

(ii) void what(int n)
{
  if (n==1) — $O(1)$
  return n; — $O(1)$
  for (int i=1; i<=n; i++) — $O(n)$
    for (int j=1; j<=n; j++) — $O(1)$
      printf("*");
      break;
}

$O(n)$

(iii) function(int n)
{
  for (int i=0; i<n; i++) — $O(n)$
    for (int j=1; j<i*i; j++) — $O(n^2)$
      if (j%i==0)

$O(n) \times O(n^2) = O(n^3)$

$j < i*i$, $i < n$, $n \times n = n^2$

Date __/__/____

Page No.

for(k=0; k<j; k++) → $O(n^2)$

printf("*");

$\therefore$ I goes to n^2 so k also goes to n^2 → $O(n^5)$

Ans $O(n^5)$

(iv) fun(int n)

{

int i=1;

while(i<n) → $O(\log_2 n)$

{

int j=n;

while(j>0) → $O(\log_2 n)$

{

j = j/2

}

i = 2*i;

}

}

$O(\log_2(\log_2 n))$

Date __/__/____ Page No.: ____

★ Asymptotic notation:

(i) Asymptotic notations are abstract notations for describing the behaviour of the algorithm and determine their rate of growth.

(ii) The worst case time complexity of algo is that input sequence, for which it takes maximum amount of time.

(iii) Best case:
It is that input sequence, for which the algorithm take minimum time.

(iv) Why we don't study average case?

Ans:- t_1 t_2 ... t_n

E_1, E_2 ---- E_n

P_1 P_2 ... P_n

$$= \sum_{i=1}^{n} P_i * t_i$$

Suppose for a certain problem, there exist i_1 to i_n different input cases having probability from P_1 to P_n, where each take time t_1, t_2 ---- t_n respectively.
So, the final avg. case time will be

$$A(n) = \sum_{i=1}^{n} P_i * t_i$$

Date / /

Page No.:

As it takes extremly large amount of time to compute this value, so it is not possible to study average case.

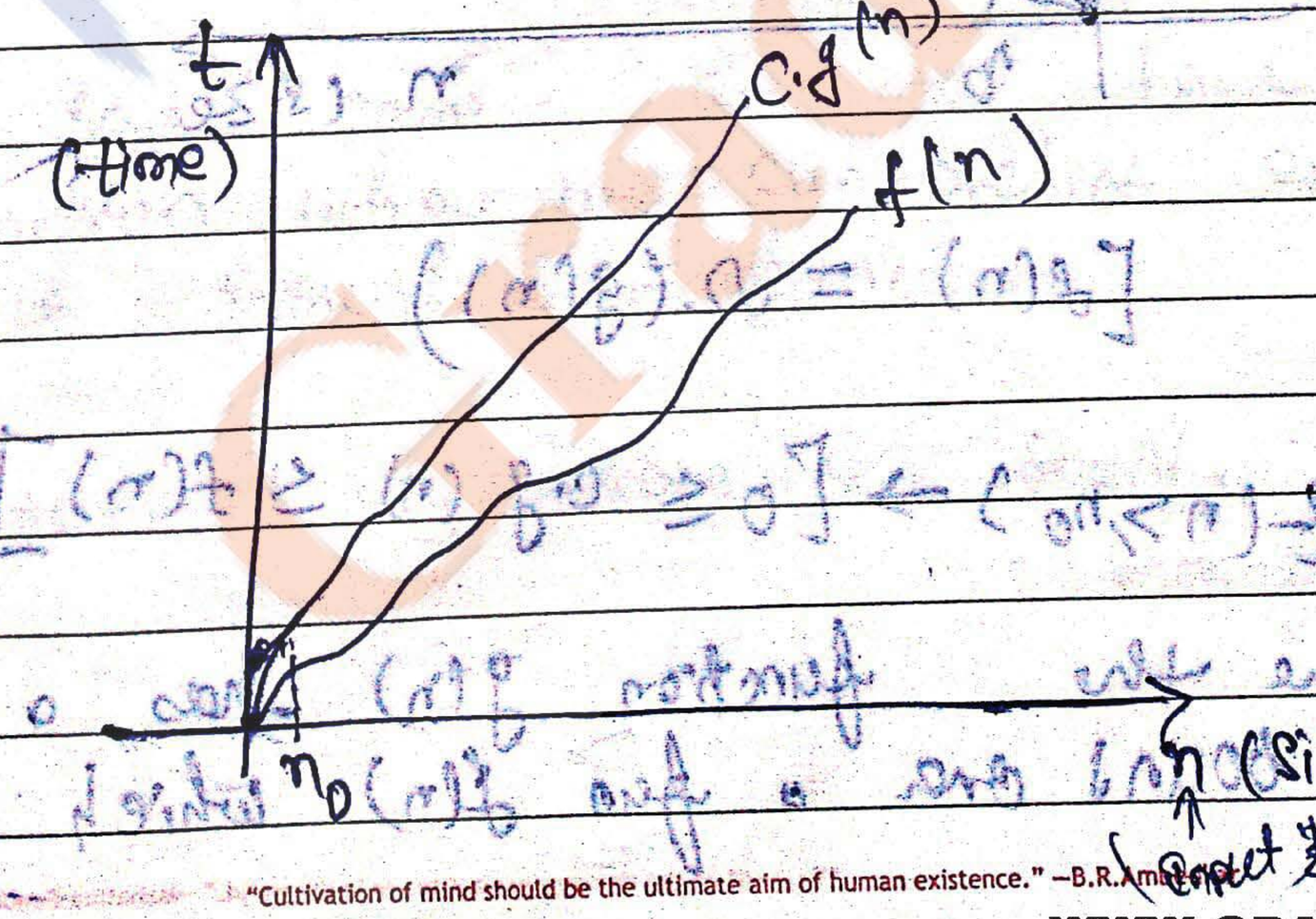
* Types

- Asymptic notation are of two types.

  - big notation
    - Big - oh (O)
    - Big - theta (Θ)
    - Big - omega (Ω)
  - small notation
    - small-oh(o)
    - small omega(ω)

(i) Big notations -

(ii) Big-oh(O) : -



Date __/__/____ Page No.: ____

By O, will always with big.

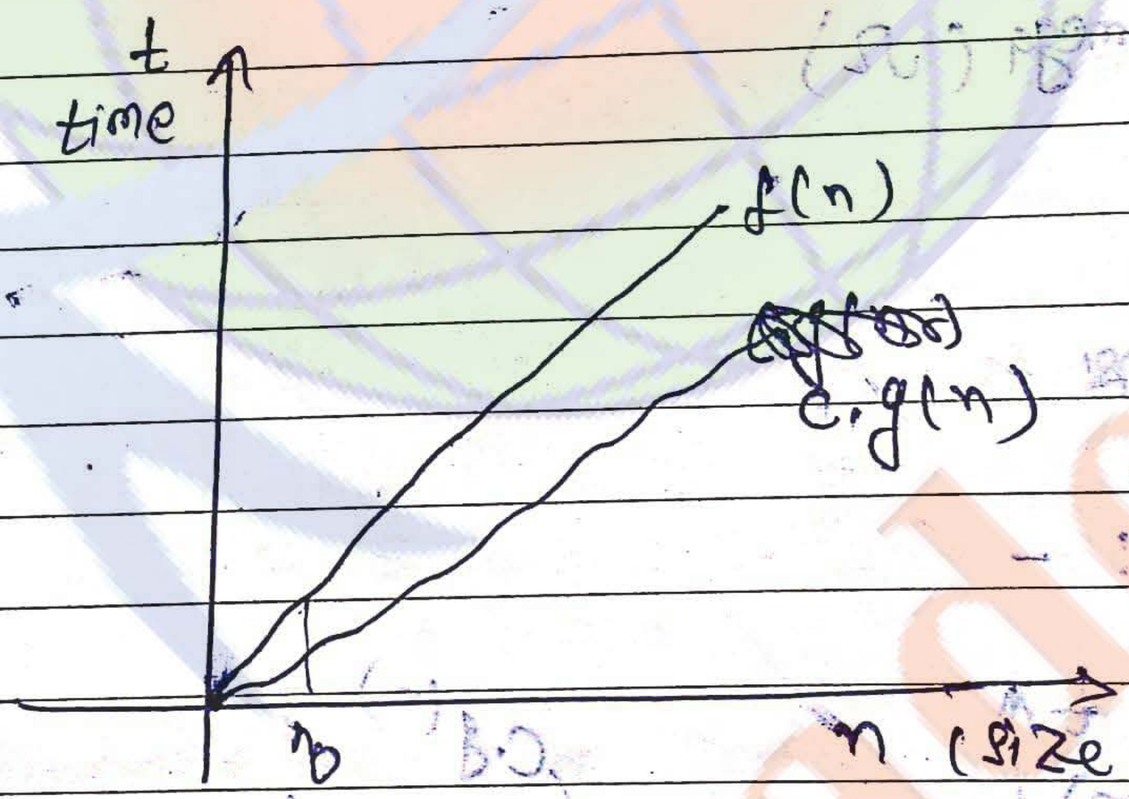
$$f(n) = O(g(n)) \rightarrow [0 \le f(n) \le c.g(n)]$$

This representation means that the upper bound of function $f(n)$ is $g(n)$ and $f(n)$ can be min to zero but always will less than $c.g(n)$. holds when $n \ge n_0$.

Q. what is c ?

Ans. c is any positive constant.

(ii) Big-omega (Ω)



$$f(n) = \Omega(g(n))$$

$$t(n \ge n_0) \rightarrow [0 \le c.g(n) \le f(n)]$$

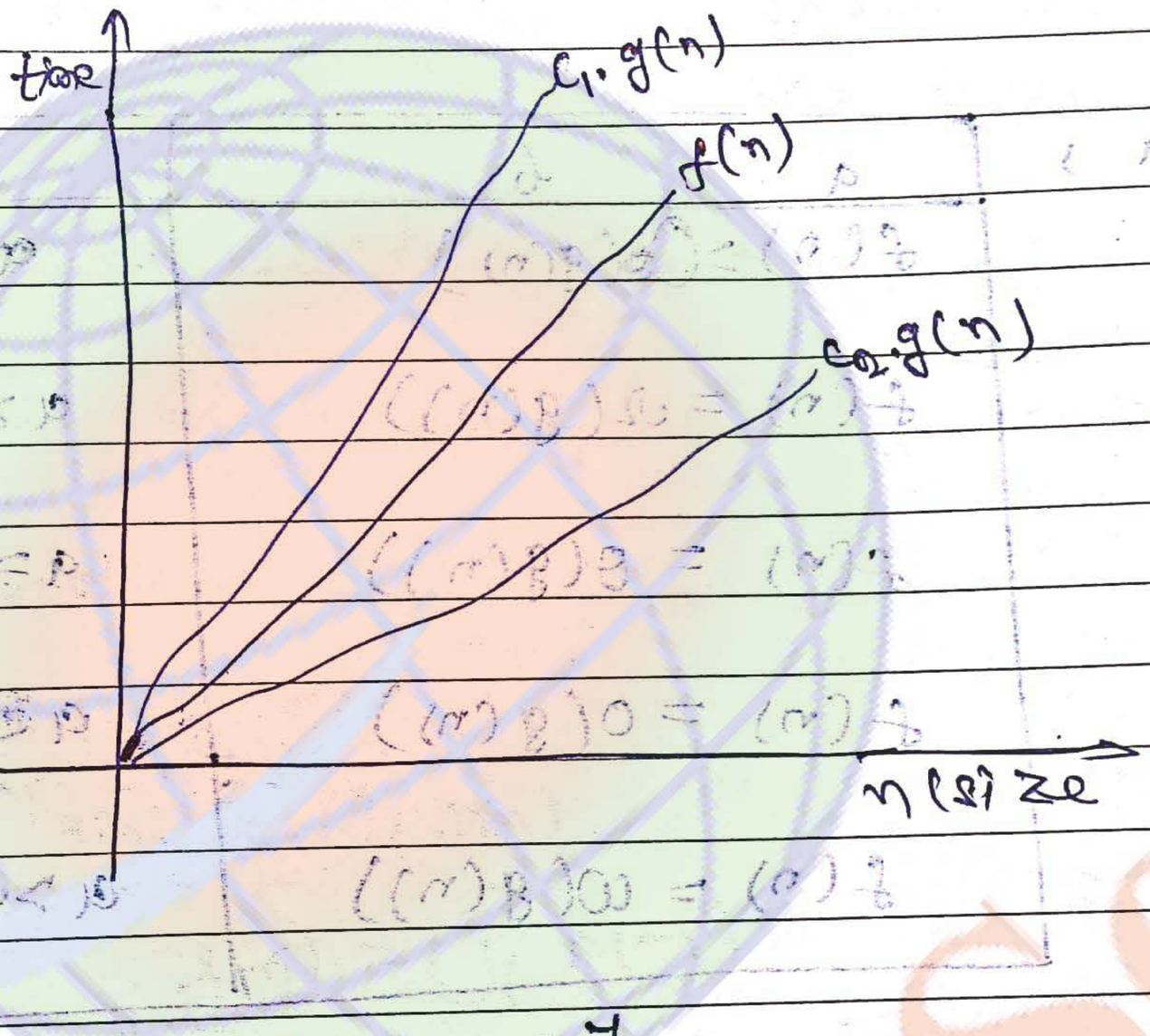
Here the function $g(n)$ has a lower bound over a fun $f(n)$ which mean's

Date __/__/____ Page No.: ____

can be min, be greater than equal to $c_1 g(n)$ where

c_1 a positive constant and satisfy in all cases where $n \geq n_0$

(iii) Big-theta(Θ) :-



$$[f(n) = \Theta(g(n)]$$

$$[0 \leq c_2 g(n) \leq f(n) \leq c_1 g(n)]$$

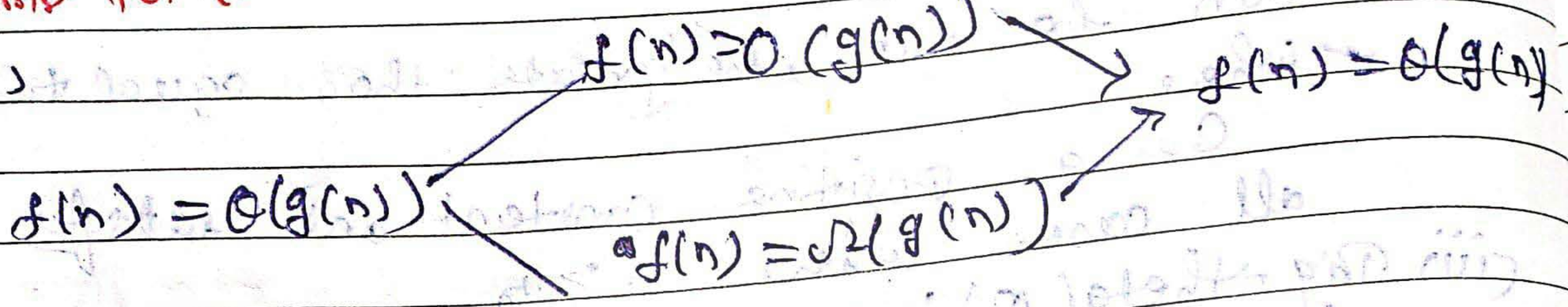
In the best case we may have a function where we can predict exactly what is growth of a function.

It is denoted by theta(Θ) bound.

Date ___/___/_____

Imp Point :-

(i)



a > b
b > c
a > c

(ii)

a	b
$f(n) = \Theta(g(n))$	$a \leq b$
$f(n) = \Omega(g(n))$	$a \geq b$
$f(n) = \Theta(g(n))$	$a = b$
$f(n) = o(g(n))$	$a < b$
$f(n) = \omega(g(n))$	$a > b$

(iii) trichotomy

In trichotomy, if there are two number then one of the following always hold good, either

$a < b$

$a > b$

$a = b$

But it is not necessary in case of function

$f(n) < g(n)$

$f(n) > g(n)$

$f(n) = g(n)$

"You must be the change you want to see in the world." –Mahatma Gandhi

Date __/__/____ Page No.: _____

eg. sino, cose)

07 Big-O satisfy reflexivity :-

$f(n) = O(f(n))$ → Big-O, Big-theta, Big-omega.

(iv) Big-notations satisfy reflexivity

(v) only theta satisfy symmetry.

(vi) Transitivity:-

$f(n) = (g(n))$

$g(n) = O(g(n)$

$g(n) = O(p(n))$

All the five notations, support's transitivity.

(vii) $f(n) = O[g(n)]$

↓

$a \cdot f(n) = O(g(n))$

If this is true then this is also true, where a is any positive constant.

(viii) If $f(n) = O(g(n))$; $p(n) = O(q(n))$

then (a) $f(n) \cdot p(n) = O(g(n) \cdot q(n))$

(b) $f(n) \cdot p(n) = O(\max[g(n), q(n)])$

Date ___/___/_____ Page No.: _____

Q> Consider two functions $g_1(n)$ and $g_2(n)$ with

$$g_1(n) = \begin{cases} n^3 & 0 \le n < 10000 \\ n^2 & n \ge 10000 \end{cases}$$

$$g_2(n) = \begin{cases} n & 0 < n \le 100 \\ n^3 & n > 100 \end{cases}$$

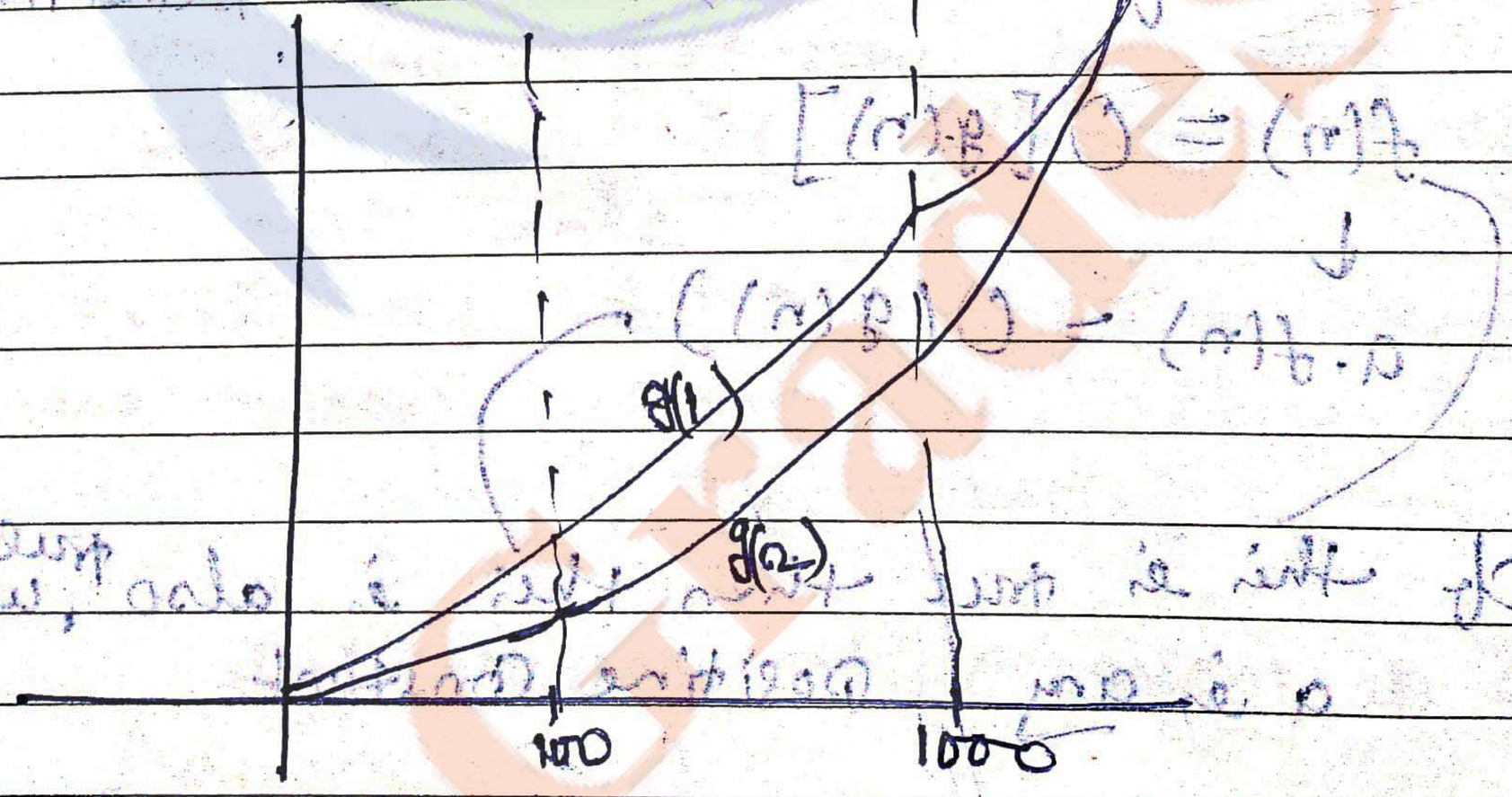
find rel^n b/w $g_1(n)$ and $g_2(n)$

(i) $O(g_1(n)) = g_2(n)$

(ii) $\Omega(g_1(n)) = \Omega g_2(n)$

(iii) $g_1(n) = \Theta(g_2(n))$

(iv) $o(g_1(n)) = g_2(n)$



Date __/__/____ Page No.: ____

gate 2008, 09

Q2) Consider the following function

$f_1(n) = 2^n$

$f_2(n) = n^{3/2}$

$f_3(n) = n \log n$

$f_4(n) = n^{\log n}$

Arrange in increasing order

Sol^n

$n = 0$

$2^0 = 1$

$1^{3/2} = 1$

$1 \log 1 = 0$

$1^{\log 1} = 1$

$f_3(n) < f_1, f_2, f_4$

$n = 10$

$2^{3/2} = \sqrt{2^3} = 1.3$

$2 \log 2 = 2 \times 0.5 = 0.6$

$2^{\log 2} = 2^{0.3} =$

$n = 1000$

$n = 10000$

$n \log n$, $n^{3/2}$, $n^{\log n}$, 2^n

polynomial — exponential

Q3) Which of the following is not true , k, m are Constant

(i) $(n+k)^m = O(n^m)$

(ii) $2^{n+1} = O(2^n)$

(iii) $2^{(2n+1)} = O(2^n)$

Sol^n false $2^{2n} = 4^n$ $O(2^n)$

true (i) $2^n \cdot 2^n = 2^{2n}$

true (ii) $(\text{const})^{\text{Const}} = (\text{Const})^{\text{Const}}$

Date ___/___/_____ Page No.: _____

Q4) $f_1(n) = n^{0.99999} \log n$

$f_2(n) = 1.00000001^n$

$f_3(n) = 1.000001^n$

$f_4(n) = n^2$

Increasing order of asymptotic notation

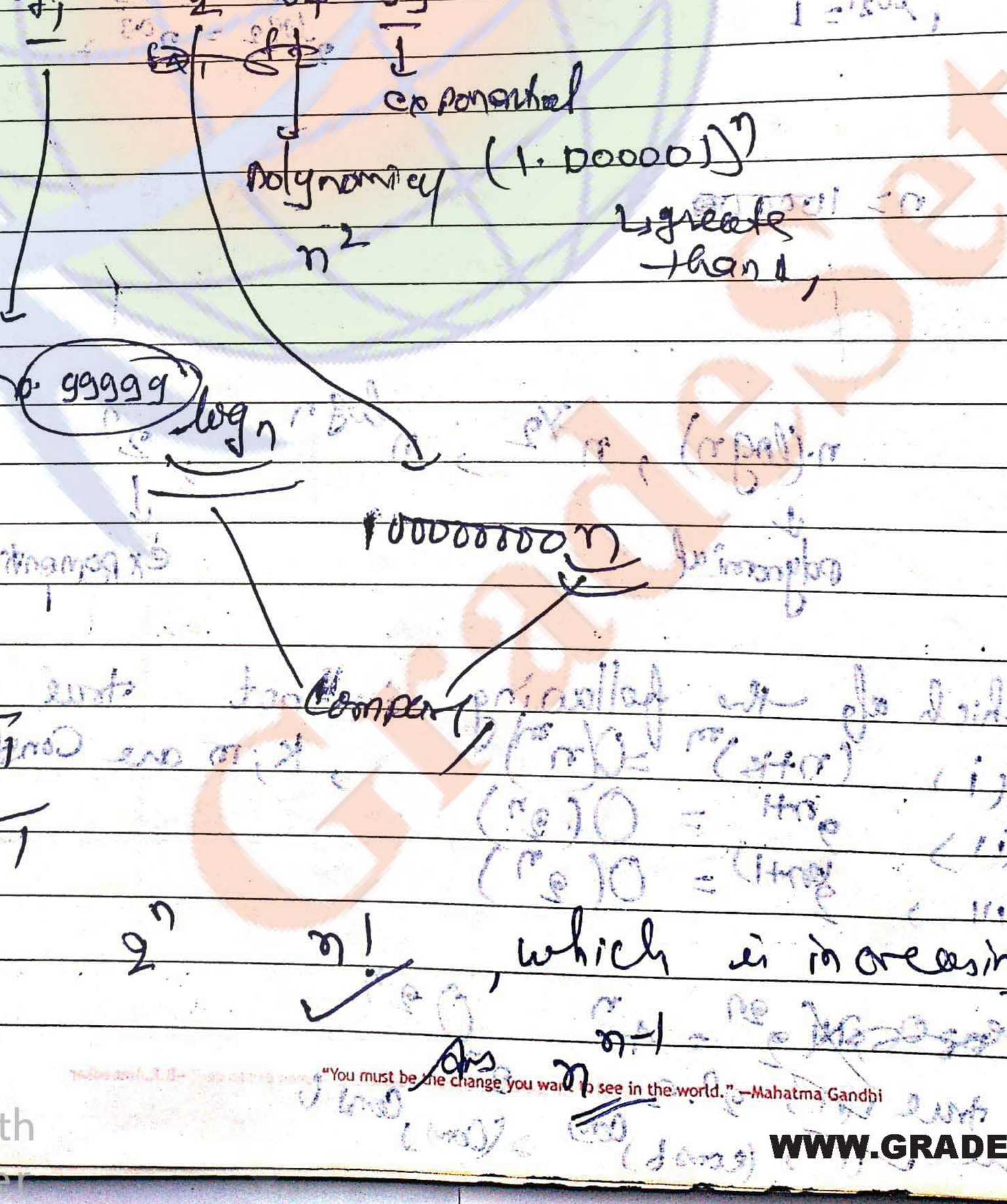
(a) f_1, f_4, f_2, f_3

(b) f_1, f_2, f_3, f_4

(c) f_2, f_1, f_4, f_3

(d) f_1, f_2, f_4, f_3

Sol:



(S) 1 2^n $n!$, which is increasing

Ans n

Date ___/___/______ Page No.: ______

(b) n^a a^n n^n

Q(7) $f(n) = 3n^{\sqrt{n}}$

$g(n) = 2^{\sqrt{n} \log_2 n}$

$h(n) = n!$

Soln

✗ (a) $h(n) = O f(n)$ — $h(n) \le f\ n$

✗ (b) $h(n) = O(g(n))$ — $h(n) \le g(n)$

✗ (c) $g(n) \neq O(f(n))$ — $g(n) \not\le f(n)$

✗ (d) $f(n) = O(g(n))$ — $f(n) \le g(n)$

Soln

$g(n)$ $f(n)$ $h(n)$

cons ~~$\sqrt{n} \log_2 n$~~ ② $n!$

Const / 2

2015

Q8) $f(n) = n$

$g(n) = n^{1+\sin n}$

where n is a +ve integer, which of the following is true.

(1) $f(n) = O(g(n))$

(11) $f(n) = \Omega(g(n))$

(a) 1 (b) 2 (c) 1, 2 (d) ~~1, 2~~

Soln

"Cultivation of mind should be the ultimate aim of human existence." —B.R.Ambedkar

$f(n) = n$ → linear growth

$g(n) \geq n^{1+\sin n}$ → show the ans -ve both

Q) Consider

$$\sum_{i=0}^{n} i^3 = X \qquad i \in Z^+$$

which is correct

(a) $\Theta(n^4)$

(b) ~~$\Theta(n^5)$~~

(c) $\Theta(n^5)$

(d) $\Omega(n^3)$

(a) 1 (b) 2 (c) 1, 3, 4, ~~2~~ (d) 2, 3, 4 ~~X~~

Sol)

$$1+2+\cdots+n = \frac{n(n+1)}{2}$$

Q) Consider four situation

(a) for (i=0 ; i<n ; i++)

(b) for (i=0 ; i<n ; i+=2)

(c) for (i=1 ; i<n ; i*=2)

(d) for (i=n ; i>1 ; i/=2)

In a competion four diff. prgm, are observes, which of the follow is most efficient

Date __/__/____ Page No.: _____

a.) Int i, n
i = 1
while (i ≤ n)
    i = i * 2

complexity = ?

(a) ⌈log n⌉ + 1

(b) n

(c) ⌈log n⌉

(d) ⌊log n⌋ + 1
→ lower bound

with n = 100

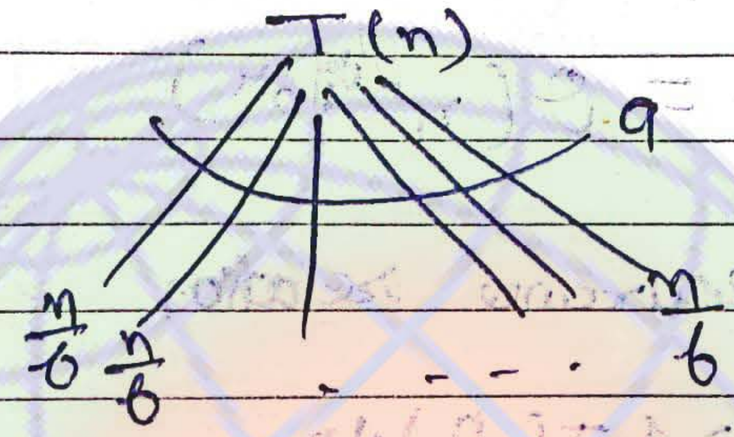
1 — 2
2 — 4
3 — 8
4 — 16
5 — 32
6 — 64
7 — 128

6 — ⌈n⌉
→ upper bound

Date __/__/____ Page No.: ____

★ Master Theorem :-

$$T(n) = a\,T\left(\frac{n}{b}\right) + f(n)$$



- It's a divide and conquer approach.

T(n)

a → $a \geq 1$

$T\left(\frac{n}{2}\right)$ → $b > 1$ → If we take 1, then we not do any work, so we need to take value greater than 1, in case of b.

- This means mean's, there is a problem of size "n" which is difficult to be solves directly so, we divides into "a" number of problems, so that each problem has a size of $\frac{n}{b}$, where f(n) is the cost of dividing the problem into subproblem and then combining these sub-soln into final soln

Date __/__/____ Page No.:

Condition! –

Condition 1st! - If $f(n) = O(n^{\log_b a - \epsilon})$ $\epsilon > 0$ (else if)

If $f(n) = O(n^{\log_b a - \epsilon})$ $\epsilon > 0$

$T(n) = \Theta(n^{\log_b a})$.

eg 1) Consider a recurrence relation

$T(n) = 4T(\frac{n}{2}) + n$

$a = 4, b = 2, f(n) = n$

$n = O(n^{\log_2 4 - \epsilon})$

$n = O(n^{2-\epsilon})$ → need to prove it

$n = O(n^{1-\epsilon})$ $\epsilon > 0$

$T(n) = \Theta(n^2)$

eg 2) $T(n) = 8T(\frac{n}{2}) + n^2$

$a = 8, b = 2, f(n) = n^2$

$n^2 = O(n^{\log_2 8 - \epsilon})$

$n^2 = O(n^{3-\epsilon})$ → needs to follow it

$\epsilon = (-2)$ → $\epsilon > 0$ को ये काम

$T(n) = \Theta(n^3)$

Date __/__/____ Page No.: _____

Q3) Consider a relⁿ

$$T(n) = 2T\left(\frac{n}{2}\right) + n \log_2 n$$

Sol^n

$$a = 2, \quad b = 2, \quad n = n \log_2 n$$

$$n \log n = O(n^{\log_2 2 - \epsilon})$$

$$n \log_2 n = O(n^{1-\epsilon})$$

$\epsilon > 0$

$\epsilon = 0$

As here condition 1 fails, so, we will go for condition 2.

↓

Condition 2nd:-

If $f(n) = \Theta(n^{\log_b a} \cdot (\log_2 n)^k) \quad k \geq 0$

$$T(n) = \Theta(n^{\log} \cdot \log n^{k+1})$$

So, Apply for above question

$$a = 2, \quad b = 2, \quad f(n) = n \log_2 n$$

$$f(n) = \Theta(n^{\log_b a} \cdot (\log n)^k)$$

$$n \log n = \Theta(n^1 \cdot (\log n)^{k=1})$$

$$T(n) = \Theta(n (\log n)^2)$$ Ans

Date ___/___/_____

$$\log_b a = \frac{\log_{10} a}{\log_{10} b}$$

Page No.: _____

Q Consider a recurance

$$T(n) = 7T\left(\frac{n}{2}\right) + n^2 \rightarrow$$

$a = 7, \quad b = 2, \quad n = n^2$

$$n^2 = O(\log_2^{7-\epsilon})$$

$$n^2 \geq O(\log_2^{7-\epsilon})$$

Not equal to possible

So, condition 1 is false,

So we need to apply condition 2

$$T(n) = O(n^{\log_2 7})$$

$$= O(n^{2.81})$$

Ans

$a \geq 7, \quad b \geq 2, \quad f(n) \neq O$

Q> $T(n) = 7T\left(\frac{n}{2}\right) + n^2$

Q>

8/7

MTech ✗ Research → ✗
Teaching → ✓

Date __/__/____ Page No.: ____

a) $T(n) = T(\frac{n}{3}) + n$

b) $a = 1, \quad b = 3, \quad f(n) = n$

$f(n) = O(n^{\log_3 1 - \epsilon})$

$n = O(n^{\log_3 1 - \epsilon})$ ✗

~~$f(n) = O(n^{\log_3 1})$~~ Condition 1 fail

Condition 2:-

$f(n) = O(n^{\log_3 1} \cdot (\log n)^k) \quad k \geq 0$

$= O(n^{\log_3 1} \cdot (\log n))$

$= O(n^{\log_3 1} \cdot 1)$

Condition 2 fails.

So, we need to see 3rd condition

Condition 3:-

If $f(n) = \Omega(n^{\log_b a + \varepsilon}) \quad \varepsilon > 0$

$\hookrightarrow a \cdot f(\frac{n}{b}) \leq c \cdot f(n) \quad c < 1$

$\hookrightarrow T(n) = \Theta(f(n))$

For we solve above,

Date ___/___/_____ Page No.: _____

$a = 1, \quad b = 3 \quad f(n) = n$

$n = \theta(n^{\log_3 1 + \epsilon})$

$1\left(\frac{n}{3}\right) \leq \epsilon \cdot n$

$\frac{1}{3} \leq \epsilon$

$T(n) = n$

Q) Consider

```
f(int n)
{
  if (n < 2)
    return n
  else
    counter = 0
    for i = 1 to 8
      f(n/2)

    for i = 1 to n^3
      Count = Count + 1
}
```

complexity of function f.

Sol n

Citra

Date __/__/__ Page No.: __

~~n<1?~~ n>1] ~~n=1, count=0,~~ f(n/2)

g =

Count = count =

(f) $T[n] = 8T\left[\frac{n}{2}\right] + n^3$

$a = 8,\ b = 2,\ f(n) = n^3$

$f(n) = O(n^{\log_2 8 - \epsilon})$

$n^3 = O(n^{3-\epsilon})$ ✗

~~f(n) = Θ(...)~~

Condition 2

$n^3 = \Theta(n^3 \cdot \log n^k)$ for k=0, it will satisfy /

k = 0

$T(n) = \Theta(n^3 \log n)$

Q.) Consider a code

void function (int n)

{

if (n ≤ 1)

return 1

"Cultivation of mind should be the ultimate aim of human existence." —B.R.Ambedkar

Date __/__/____ Page No.: ____

```
if (n>1)
{
  print ("*");
  function (n/2);
  function (n/2);
}
```

Soln $T(n) = 2T\left(\frac{n}{2}\right) + 1$

$a = 2, \quad b = 2, \quad f(n) = 1$

$1 = O\left(n^{\log_2^2 - \epsilon}\right)$

$= n^{1-\epsilon}$

$n^0 = n^0$ if $\epsilon \geq 0$

$T(n) = \theta\left(n^1\right)$

Q.) Let An represent the no of bit strings of length n, containing two consecutive ones, what is the correct recurrence rel for a^n.

(a) $a_{n-2} + a_{n-1} = 2^n$

(b) $a_{n-2} + 2a_{n-1} = 2^{n-2}$

(c) $2a_{n-2} + a_{n-1} = 2^{n-2}$

(d) $2a_{n-2} + 2a_{n-1} = 2^{n-2}$

Date __/__/____ Page No.: ____

Q/7

$q_0 \rightarrow 0$

$q_1 \rightarrow 0$

$q_2 \rightarrow 1$

$q_3 \rightarrow 1$ 000

$q_4 \rightarrow 0$ 001

010

011

100

101

110

111

0	0000
1	00001
2	0010
3	0011
4	00100
5	0101
6	0110
7	0111
8	

for $n \geq 4$

$q_4 \rightarrow q_{4-2} + q_{4-1} + 2^{n-2}$

$q_2 \rightarrow q_0 + q_1 + 2^2$

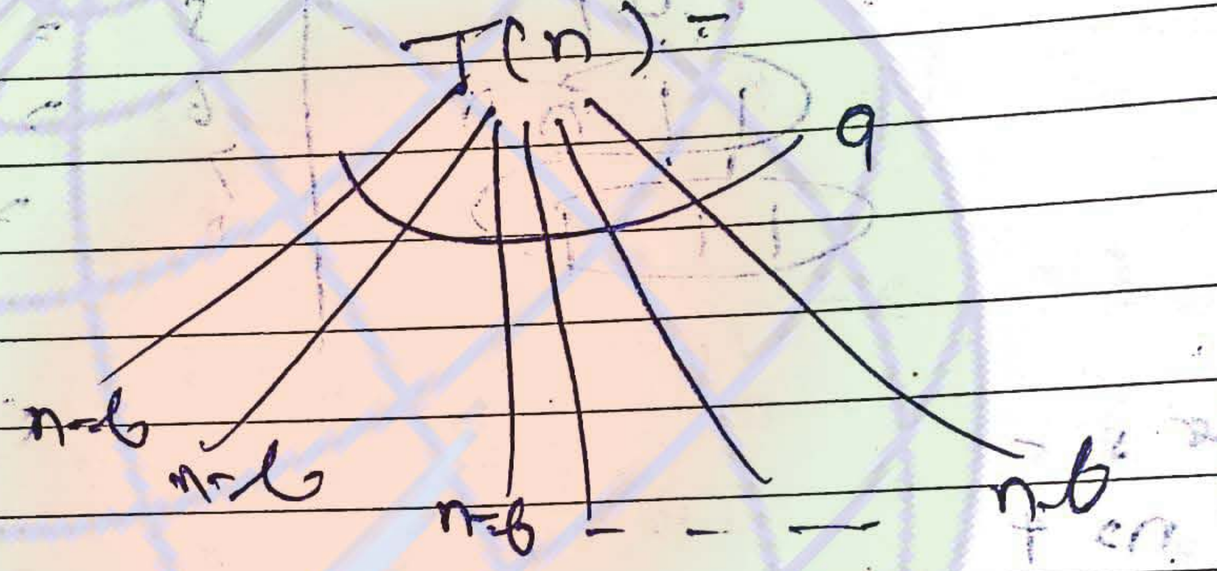
Date __/__/____ Page No.: ____

☆ Substract and Concour:-

$$T(n) = aT[n-b] + f(n)$$

$a \geq 1$

$b > 0$



This is standard substract and concour recurance, where a problem of size "n" is divided into "a" number of problems each of size n-b.

If $f(n) = O(n^k)^{k \geq 0}$

↳ If $a = 1 \rightarrow O(n^{k+1})$

If $a > 1 \rightarrow O(n^k a^{n/b})$

Q1) $T(n) = T(n-1) + n$

Q2) $T(n) = 3T(n-1) + 4$ Ans 3^n

Date __/__/____ Page No.: ____

(03) $T(n) = T(n-3) + 2n^2$ → n^3

(04) $T(n) = T(n-1) + n^2$ → n^3

Q/7

(i) $a = 1$, $b = 1$, $f(n) = n$

$n = O(n^k)$ $k = 1$

↳ If $a = 1$ → $O(n^2)$

(ii) $a = 3$, $b = 1$, $f(n) = 1$

$n = O(n^k)$

$k \geq 1$

↳ If $a > 1$

Date ___/___/_____ Page No.: _____

standard logic/trick:-

* $T(n) = T(\alpha n) + T((1-\alpha)n) + \beta n$

$0 < \alpha < 1$

$\beta > 0$

$T(\frac{n}{3}) + T(\frac{2n}{3})$ → $\frac{n}{3} + \frac{2n}{3} = n$

↳ $O(n \log n)$ — Substitute then

Q) Con

Scoping and calling methods

Date Page No.

If not say anything then call by value

Q7 Consider a following code

```
var x, y : int
Procedure P(n: int)
Begin
    x = (n+2)/(n-3)
end
Procedure Q
var x, y : int
begin
    x = 3
    y = 4
    P(y)
    write(x)
end
begin
    x = 7
    y = 8
    Q
    write(x)
end
```

Soln Static scoping = 3, 6

Dynamic scoping = 6, 7

Q In a function we require to access a variable either to store or to read then in case of static scoping we will come out Bracket by bracket and will search the required variable while

Date __/__/____ Page No.:

In case of dynamic scoping, we will search the function by which the current fn' is call and will repeat the procedure.

a) Program P2
var n:int
Procedure x(var x:int)
begin
x = x+1
Print x
end
Procedure D
begin
var n:int
n=3
x(n)
end
begin
n=10
D
end

n=3
x=n+1 =4
Print x → '4'

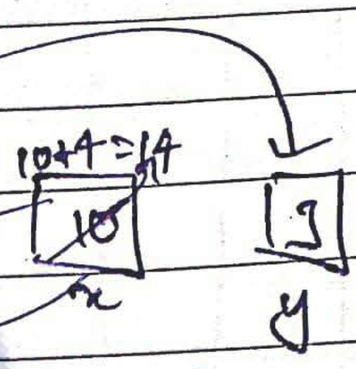
3 n
reference

Case 1: Call by value, static scoping
Ans: 11
Static scope
Dynamic scope
call by reference =

Date __/__/____ Page No.: _____

Q.7 Consider a function

```
Program P1( )
{
    x = 10
    y = 3
    fun(y, x, x)
    Print x
    Print y
}
fun(x, y, z)
{
    y = y + 4
    z = x + y + 3
}
```



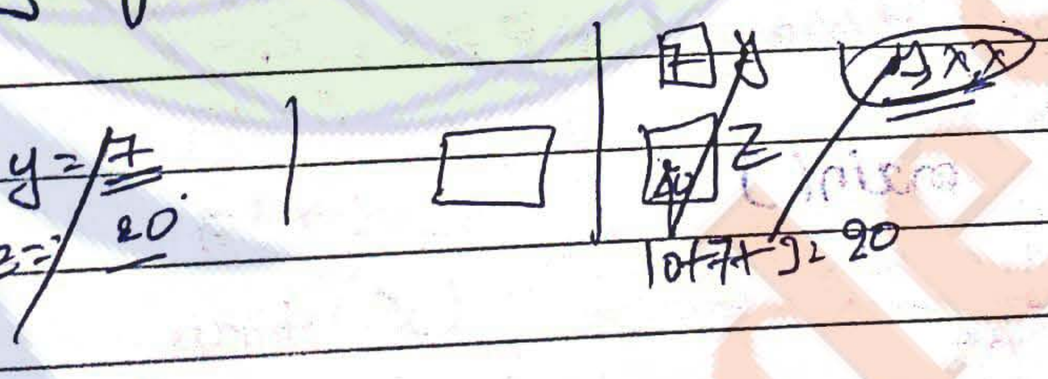
⇒ 10+4=14

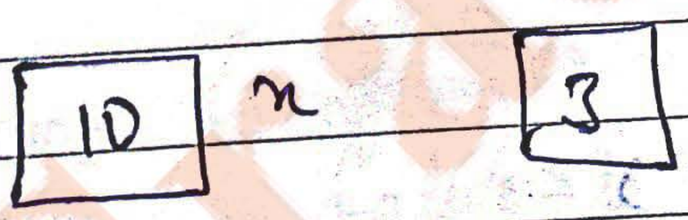
⇒ 14+3+3 = 20

Call by value = ?

Call by reference = ? ⇒ 20, 3 Ans

Sol:





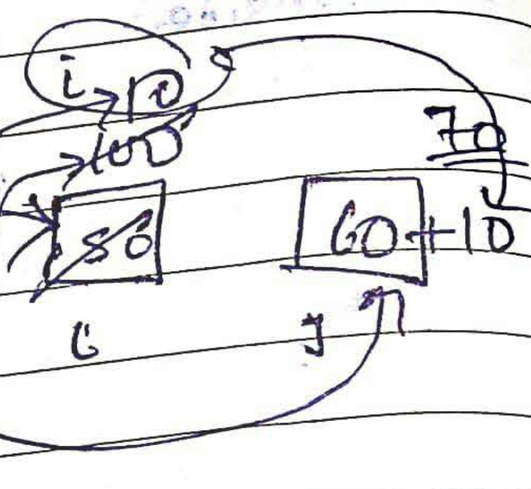
Q) Consider a Code

```
int i
Program main( )
{
    int y = 60
```

Date ___/___/_____ Page No.: _____

call by value

```
i = 50
Call f(i,j)
Print i, j
}
void f(x,y)
{
i = 100
x = 10
y = y + i
}
```



Call by value = 50, 60

Call by reference = 10, 70

```
int i
Program main()
{
  i=10
  Call f()
}
Procedure f()
{
  int i = 20
  Call g()
}
Procedure g()
{
  Print i
}
```

T = global i ?,

Date __/__/____ Page No.: ____

70

60 + 10

so|n Dynamic = 20

static = 10

Q.) Program main()

Static | Static | Dyna

var x,y: int

Procedure Q (Z: int) 6

begin

Z = Z + x = 6+5 = 11

write(Z)

end

static - 4 6

Procedure P(x: int) → 6

x → 6

begin

x = y + 2

Q (x)

write(x)

end

begin

x = 5, y = 4

P(x)

Q(x)

write(x)

end

SS-V → 11, 6, 10, 5

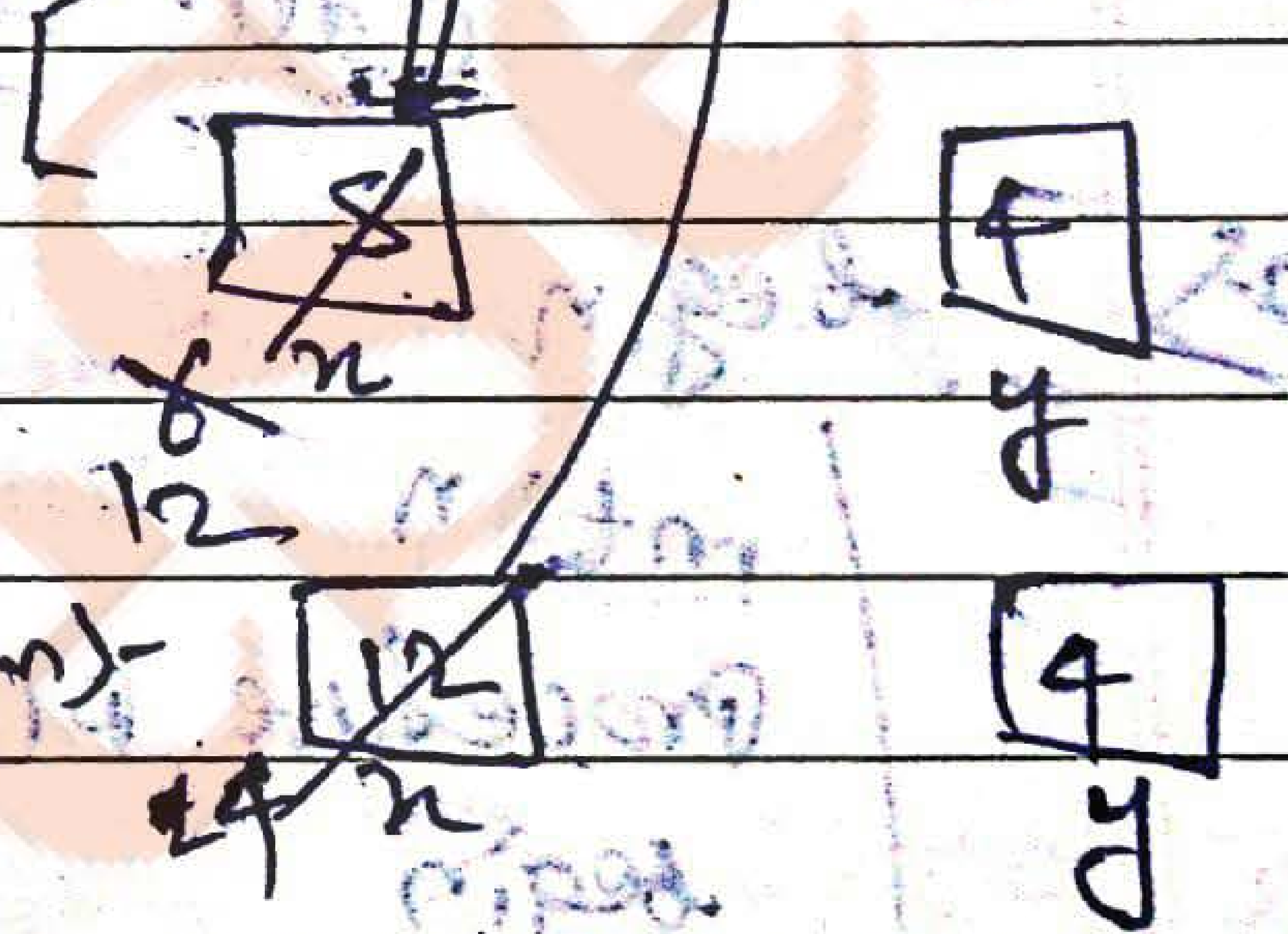
SS-R [12, 12, 24, 24

DS V 12, 6, 10, 5

DS R - 12, 12, 24, 24

p(x)

Q(x) -



Ans: I

Static scoping by value = ? 12, 12, 24, 24

State scoping by value = ? 11, 6, 10, 5

Dynamic scoping by reference = ?

State " " reference = ?

so|n

x = 6, 8 | 6

Q(Z: int)

Date __/__/____ Page No.: ____

☆ Laisy evalation technique (Call by name & Call by need) → In case of expression

a) Consider a function and find the function using call by value

```
begin
int: n
Procedure P(k: int) → k=n+10
begin
  Print(k)
  n=n+1
  Print(k)
end
n=0
P(n+10)
end
```

soln call by value = 10, 10
call by name / laisy evalation = 10, 11
call by need = 10, 11

a)

```
begin
int: n
Procedure P(k:int → P=n/10
begin
Print(n)
end
n=5
P(n/0)
end
```

soln call by value = X (runtime error)
call by name = 5

Date __/__/____ Page No.: _____

Imp.

(i) call by name and call by need are lazy evaluation technique.

where

If a expression is passed then instead of solving it, we will pass the expression as it is.

(ii) Call by name will evaluate the expression every time

while

Call by need will evaluate the expression one time.

Q.) Consider a code and find the output using call by name and call by need.

global int i=100, j=5

void P(x) → x=i+j

10+5=15

{

int i=10

Print (x+10) → 25, 15

i=200

j=20

print (x) → i+j

}

main

{

P(i+j)

}

i=100, j=5

10

→ 205

→ 220

20, 5, 2022

Static scoping call by need: 25, 15

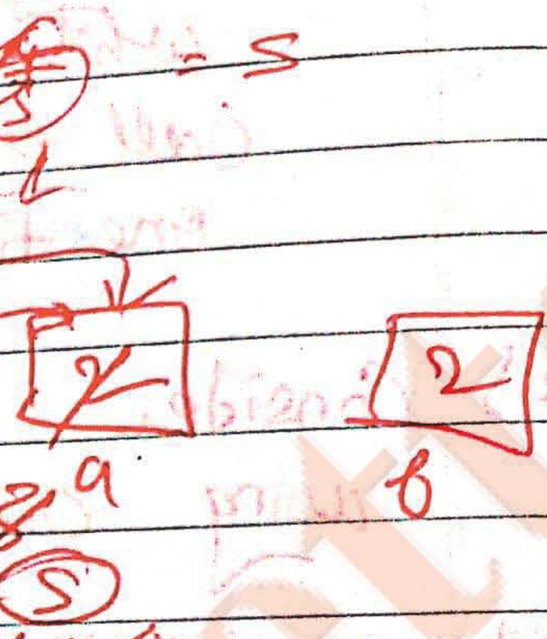
Dynamic scoping call by name: 25, 220 Ans

Date __/__/____ Page No.: ____

Call by value store/copy store!—

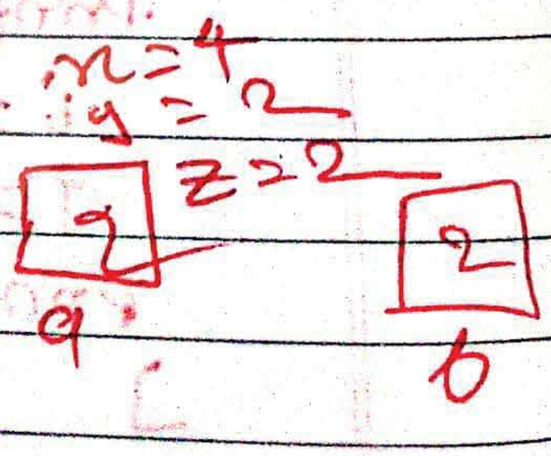
Q) Consider a code,

```
Procedure p(x, y, z)
begin
    y = y+1
    z = x + x/y
end
begin main( )
{
    a = b = 2
    p (a, b)
    Print(a)
}
```



Call by value = 2
Call by reference = 5

CV ⇒ a+b = 4, 2 2
4, 2



Call by copy store — Both 3 or 5

Date / /

Page No.:

Human body

Here we call a function like call by value but at a time of termination, we will store the value in the calling function.

★ Static Variable :-

eg:

```
int f (int n)
{
    static int i=1
    if (n>=5)
        return n
    n=n+1
    i++
    return f(n)
}
```

f(1)

(a) 5 (b) 6 (c) 7 (d) 8

sol: f(1)

f(2)

i=3

n=2

i=2

7

2012 Q.

```
int a, b, c=0
void fun (void);
int main()
{
```

China

Date ___/___/_____ Page No.: _____

```
static int a=1
Print fun();
  a=a+1
  Printfun();
  Print("%d %d", a, B);
}
```

```
void Printfun(void)
{
static int a=4 4 / 4
int b=2
a=a++b
print("%d %d", a, b)
}
```

(a) 1 4 2

(b) 4 2 6 1 6 1

(c) 4 2 6 2 20

(d) 1 4 52 52

```
Q) int incr(int i)
{
static int count=0 1 3 6 10
count = count++i;
return (count)
}
```

Date __/__/____ Page No.: ____

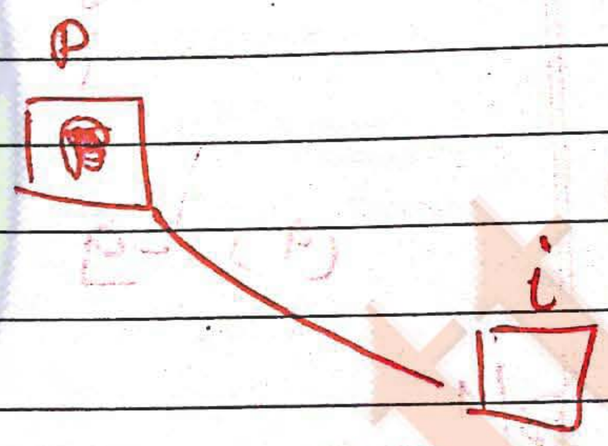
```
main()
{
  int i,j
  for(i=0; i<=4; i++)
      j = incr(i)
}
```

0 1 2 3 4

what is the value of j at the end,

(a) 0 (b) 4 (c) 6 (d) 7

a>
```
# include<stdio.h>
main()
{
  int i;
  int *p = &i;
  scanf("%d", p)
  printf("%d", i+5)
}
```



(a) Compilation error (b) runtime err.

(c) (d) ~~content~~ more than,

solved

a>
```
include<stdio.h>
intf(int *a, int n)
{ if (n<=0)
    return 0
```

Page No.:

Date

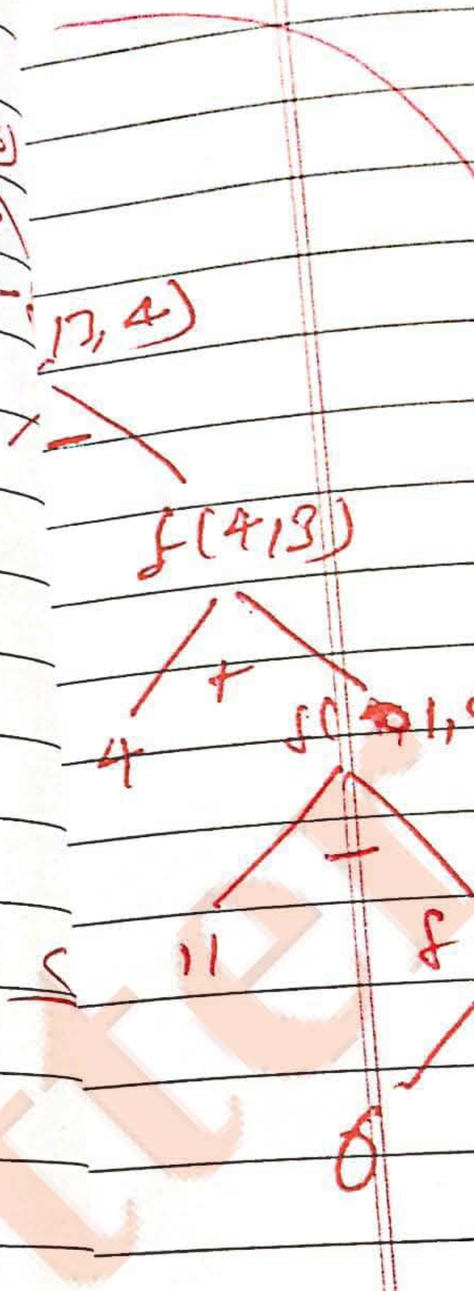
```
else if (a%2==0)
    return a+f(a+1, n-1)
else
    return a-f(a+1, n-1)

int main()
{
int a[] = {12, 7, 13, 4, 11, 6};
printf("%d", f(9,6));
getch();
}
```

(9) -9 (6) 5 (5) 15, (d) 19

Ans

```
int f(int a, int b)
{
if(a%2==0)
    return a+f(a+1, b-1);
else
    return a-f(a+1, b-1);
}
```




```
Q> include <stdio.h>
int fun(int n, int *fp)
{
  int f ff;
  if(n<=1)
  {
    *fp=1
    return 1
  }
  t= fun(n-1, fp)
  f=t+*fp
  *fp=t
```

Date __/__/____ Page No.: _____

```
return 0
int f
}
int main()
{
int x=15
printf("%d", fun(5,&x));
return 0
}
```

50/4

(a) 6 (b) 9 (c) 14 (d) 15

```
int fun(int n, int *fp)
{
int f;
*fp = n+4;
f = fun(n-1, fp);
return f*f;
}
```